HOCHSCHULE BOCHUM

Dokumentation der Projektarbeit im Fach Industrieroboter

(Gruppe: Wibke Lorenz, Mahmood Shubbak, Peng Xu)

Inhaltsverzeichnis

1. Allgemeine Aufgabenstellung

Im Rahmen der Praktika im Fach Industrieroboter wird eine Projektarbeit mit Industrieroboter durchgeführt. Bei der Aufgabe handelt es sich um eine Handhabungsanwendung inklusive Sortieraufgabe (pick and place).

Die vom Professor vorgegebenen allgemeinen Randbedingungen lauten:

- Die zu entwickelnde Anwendung benutzt das Werkzeugkoordinatensystem (TCP, UTOOL) und ein Anwenderkoordinatensystem (UFRAME)
- Mit Hilfe der am Greifer vorhandenen Sensoren ist zu prüfen:
  - ob der Greifer vor dem Greifen eines Teiles geöffnet ist,
  - ob das Teil nach dem Greifen sicher gegriffen ist und
  - ob das Teil vor dem Ablegen noch im Greifer ist.

  Hinweis: Diese Prüfungen sind an allen notwendigen Stellen einzubauen. Bei allen Sensorabfragen ist für den Fehlerfall eine Reaktion des Programms **und** des Roboters vorzusehen. Dabei sollte insbesondere berücksichtigt werden, dass ein Sensor auch fehlerhafte Informationen liefern kann.
  Ziel sollte es immer sein, bei einem Fehlerfall möglichst die Anwendung nach einem Bedienereingriff fortzusetzen (möglichst kein Abbruch).
- Das Programm muss so gestaltet werden, dass bei der Ausführung der „General Override“ auf **100%** steht.
- Die Anwendung muss ohne Reduzierung der Geschwindigkeit **sicher** funktionieren.
- Die Anwendung muss in möglichst kurzer Zeit die Sortieraufgabe erledigen (time is money).
- Es kann bei dem Sortiervorgang **nicht** davon ausgegangen werde, dass die Werte des Unterscheidungsmerkmals für die Teile a priori bekannt sind. Jedes Teil muss gemessen und seinem Messwert (für Gewicht, Länge oder Durchmesser…) entsprechend in die Gesamtmenge der Teile eingegliedert werden.
- Einschalttaster und Meldelampen müssen verwendet werden,
  weiß = Quittierung erforderlich, grün = Automatikbetrieb,
  gelb = Bedienereingriff erforderlich, rot = NOT-AUS (Fault-Zustand, erlischt bei RESET)
- Im Fehlerfall die Betriebsanzeige auf TP umschalten und Bildschirm für Bedienereingriff anzeigen.
- Das Programm muss auf „NOT-AUS“ reagieren (rote Lampe an).
  NOT-AUS Error Codes: SOP = 11001, TP = 11002, Ext = 11007, Fance = 11038
  Tasten: Fault Reset = OPIN[1], Reset = [153].

1.2 Ausgangssituation

Die Vorgabe der Hardware der Projekte, ist zugleich die Vorgabe der einzelnen Sortieraufgaben. Unsere Sortieraufgabe besteht darin, Teile nach ihrer Länge zu sortieren.
Die uns zur Verfügung stehenden Mittel sind:

- Fanuc LR Mate 200*i* Industrieroboter
- Arbeitsbereich
- Werkstücke
- Computer

1.2.1 Der Industrieroboter

Der Industrieroboter ist ein Roboter der Firma Fanuc und gehört zu der Serie LR Mate 200*i.* Er ist ein 6-achsiger Roboterarm, der durch elektrische Antriebe bewegt wird.
Wichtige Kenndaten sind:

- Tragfähigkeit von 3kg
- Wiederholgenauigkeit von +/- .04mm (+/- 0.002”)
- Achsengeschwindigkeit bis zu 480° in der Sekunde
- R-J3 Steuerung

1.2.2 Der Arbeitsbereich

Der Arbeitsbereich besteht aus:

- einer Aufnahmestation (**S**ource Positions S[i])
- einer Ablagestation (**D**estenation Positions D[i])
- einem Sensor
- den Signallampen
- einem Bedienerfeld mit einem NOT-AUS-Schalter, einem Start-Taster und einer weißen Signallampe
  (siehe Bild 1)

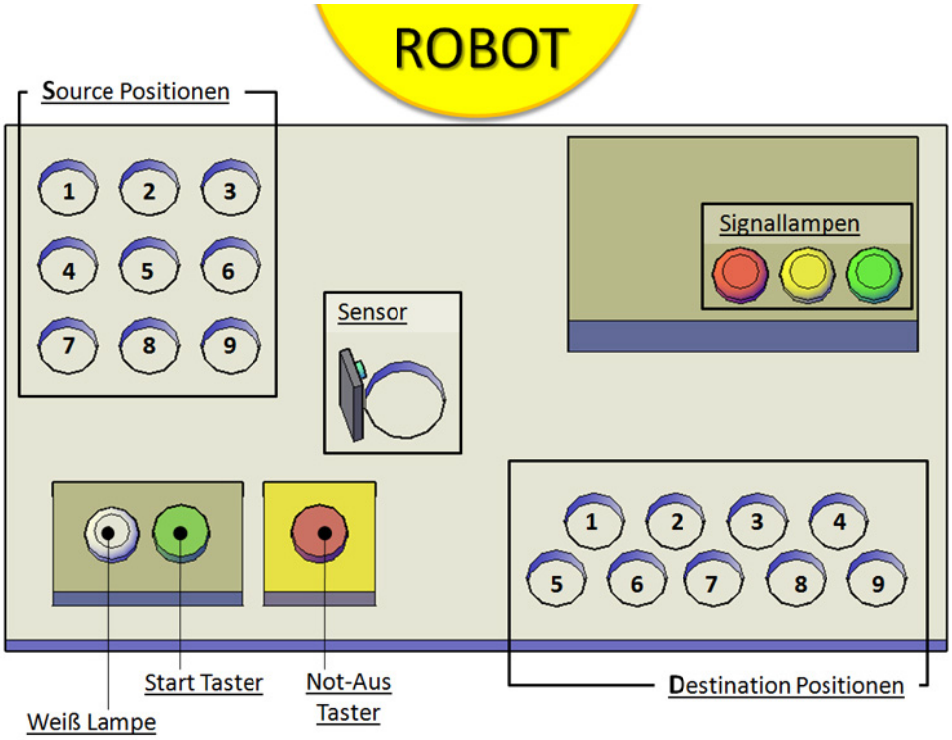


Bild 1

1.2.3 Die Werkstücke

Zur Verfügung stehen uns 9 zylindrische Werkstücke, die nach ihrer Länge sortiert werden sollen. (siehe Bild 2)

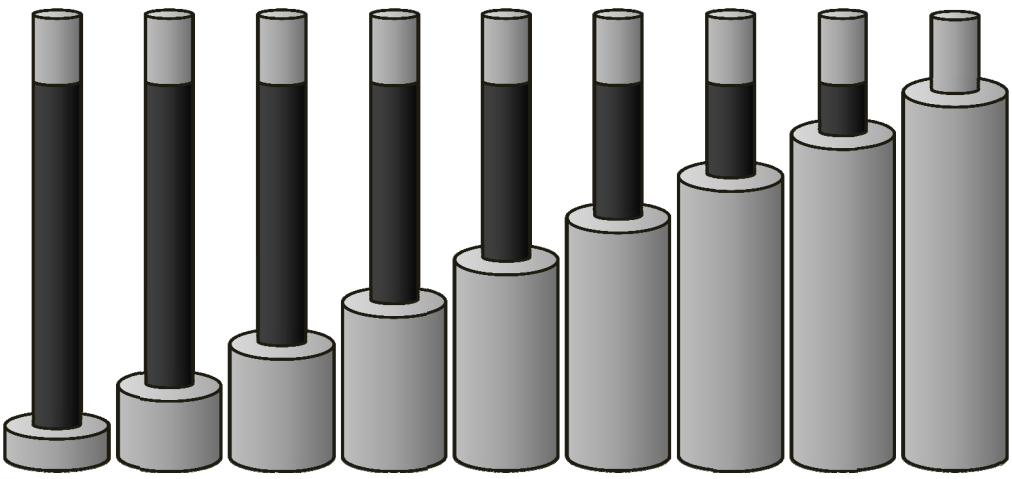


Bild 2

2. Die Sortieraufgabe

Die Aufgabe des Roboters besteht darin, Teile nach ihrer Länge zu sortieren.
Der Roboter fährt dazu zuerst zur Entnahmestation, greift ein Teil und misst die Länge des Werkstücks an einem Sensor. Wenn er alle Teile durchgemessen hat, sortiert er die Werkstücke nach der Länge geordnet in die Ablagestation ein und begibt sich danach in seine Grundstellung zurück.

2.1 Der Messvorgang

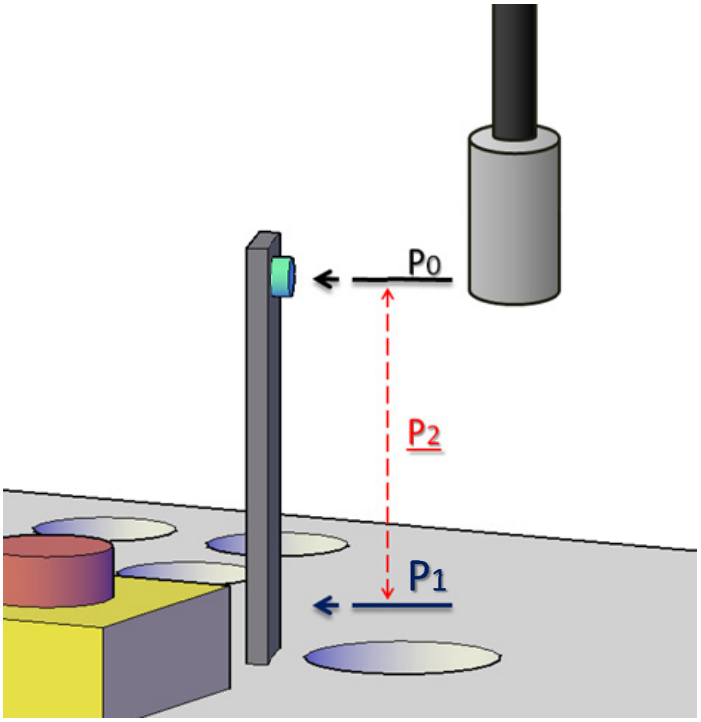
Der Messvorgang kann auf zwei unterschiedliche Art und Weisen ausgeführt werden.
Die 1. Möglichkeit ist:

- Der Messvorgang startet bei P0
- Der Sensor hat ein Signal
- Der Roboter fährt linear zu P1
- Sobald der Sensor kein Signal hat, wird die Position in P2 gespeichert.
  - **Länge = P0 – P2**

Die 2. Möglichkeit ist:

- Der Messvorgang startet bei P0
- Der Sensor hat ein Signal
- Der Roboter fährt linear zu P1
- Sobald der Sensor kein Signal hat, wird die Position in P2 gespeichert und der Roboter stoppt sofort seine Bewegung
  - **Länge = P0 – P2**

Die 2. Möglichkeit bringt uns eine große Zeiteinsparung. Dies ist der Hauptgrund, weshalb wir uns für diese Möglichkeit entschieden haben.



2.2 Verhalten bei Fehlermeldungen

Eine Fehlermeldung liegt dann vor, wenn der Roboter einen Signaleingang erwartet, aber kein Signal bekommt. Dies geschieht beispielsweise, wenn er versucht ein Teil zugreifen, jedoch gar kein Teil vorhanden ist. In diesem Falle stoppt der Roboter und es wird eine Meldung auf dem Teach Pendant ausgegeben:
„Error!!
Teil nicht vorhanden“
Der Bediener hat nun die Möglichkeit zwischen 3 Funktionen zu wählen:
Funktion → Try again: Der Roboter wiederholt den Greifvorgang
Funktion → Ignor: Der Roboter ignoriert, dass er kein Teil hat und macht in seinem Bewegungsablauf weiter.
Funktion → Cancel: Der Roboter bricht das Programm ab und fährt in seiner Grundstellung zurück.

Ein weiterer Fehlerfall wäre, wenn der Roboter während der Fahrt ein Teil verliert. Sobald er an einer Stelle angekommen ist, wo er eine Signalkontrolle macht, stoppt er seine Bewegung und gibt eine neue Nachricht auf dem Teach Pendant aus:
„Error!!
Teil verloren“
Nun kann der Bediener zwischen 2 Funktionen über das weitere Vorgehen entscheiden:
Funktion → Ignor: Der Roboter ignoriert, dass er kein Teil hat und macht in seinem Bewegungsablauf weiter.
Funktion → Cancel: Der Roboter bricht das Programm ab und fährt in seiner Grundstellung zurück.

2.3 Verhalten bei Störungen

Eine Störung liegt dann vor, wenn ein oder mehrere NOT-Ausschalter betätigt wurden. In diesem Falle weist das Teach Pendant den Bediener daraufhin, dass er die Störquelle beseitigen soll, weshalb der NOT-AUS betätigt wurde. Zum weiter Fahren muss der Bediener einen der beiden Reset-Taster (einen auf dem TP, der andere auf der Steuerung) betätigen und mit Cycle-Start, das Programm weiter laufen lassen.
Original Meldung auf dem Teach Pendant:

„Not-AUS wurde betätigt
Nach Fehlerbehebung Not- Aus lösen und
Ein Reset Taster drücken und starten mit Cycle-Start!
Achtung Programm läuft weiter!“

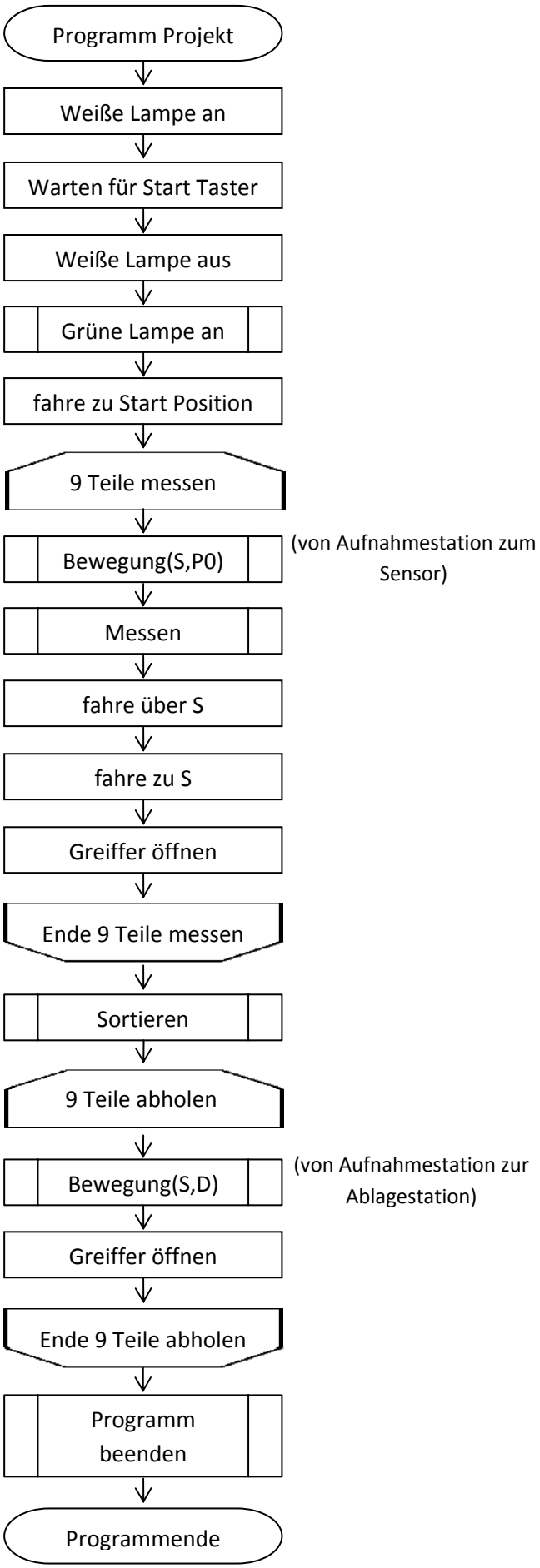
2.4 Signal-Lampen

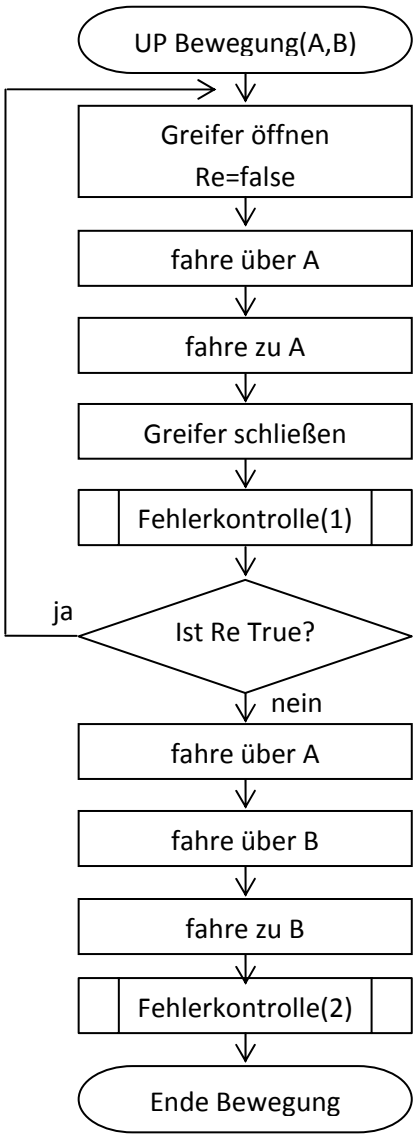
An der Anlage befinden sich vier verschieden farbige Signal-Lampen, welche den aktuellen Zustand signalisieren.

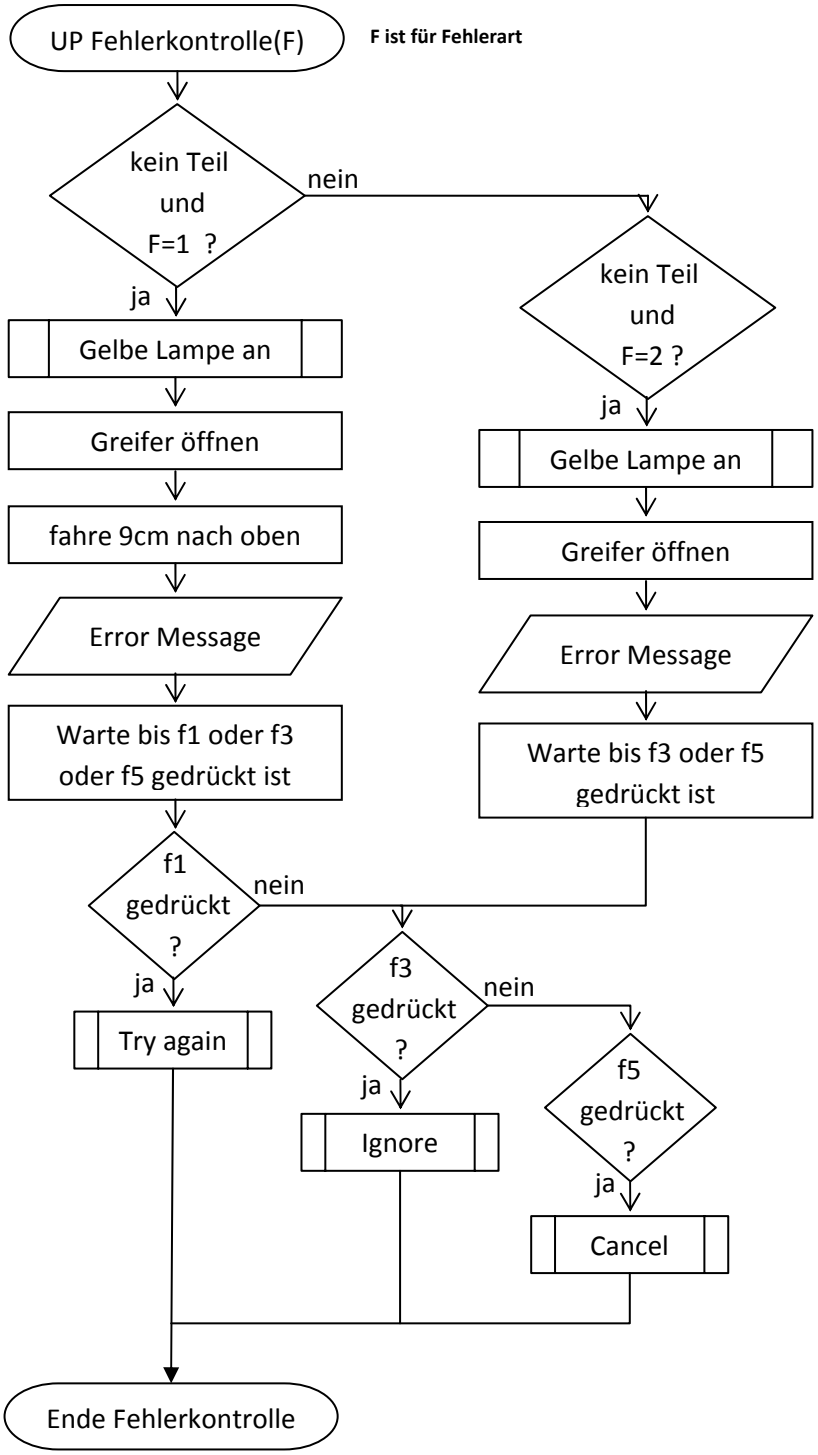
- Weiße Lampe: Programm kann gestartet werden
- Grüne Lampe: Automatikbetrieb, es liegt kein Fehler vor
- Gelbe Lampe: Fehler, kein Teil vorhanden
- Rote Lampe: Störung, Not-Ausschalter betätigt

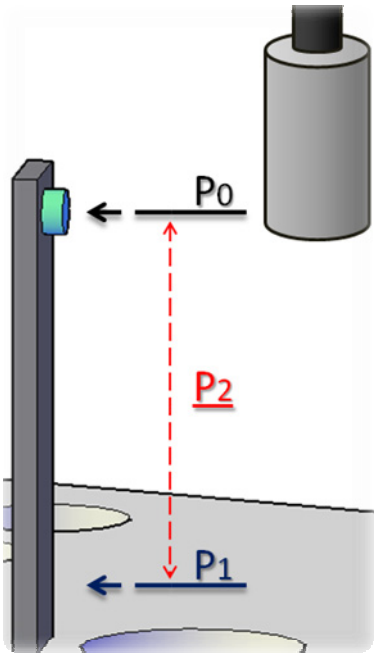
3. Das Programm

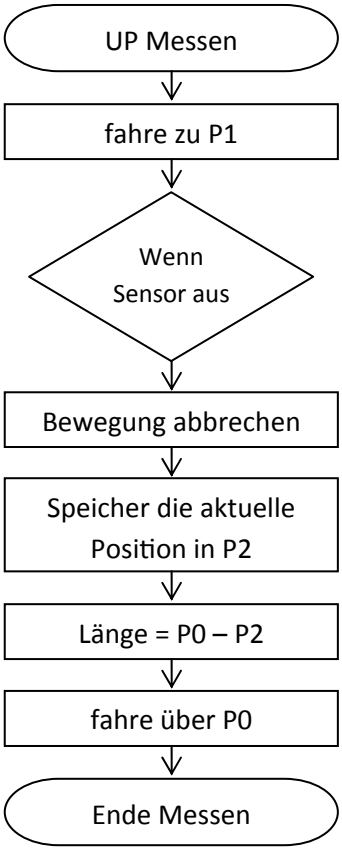
3.1 Programmablaufplan

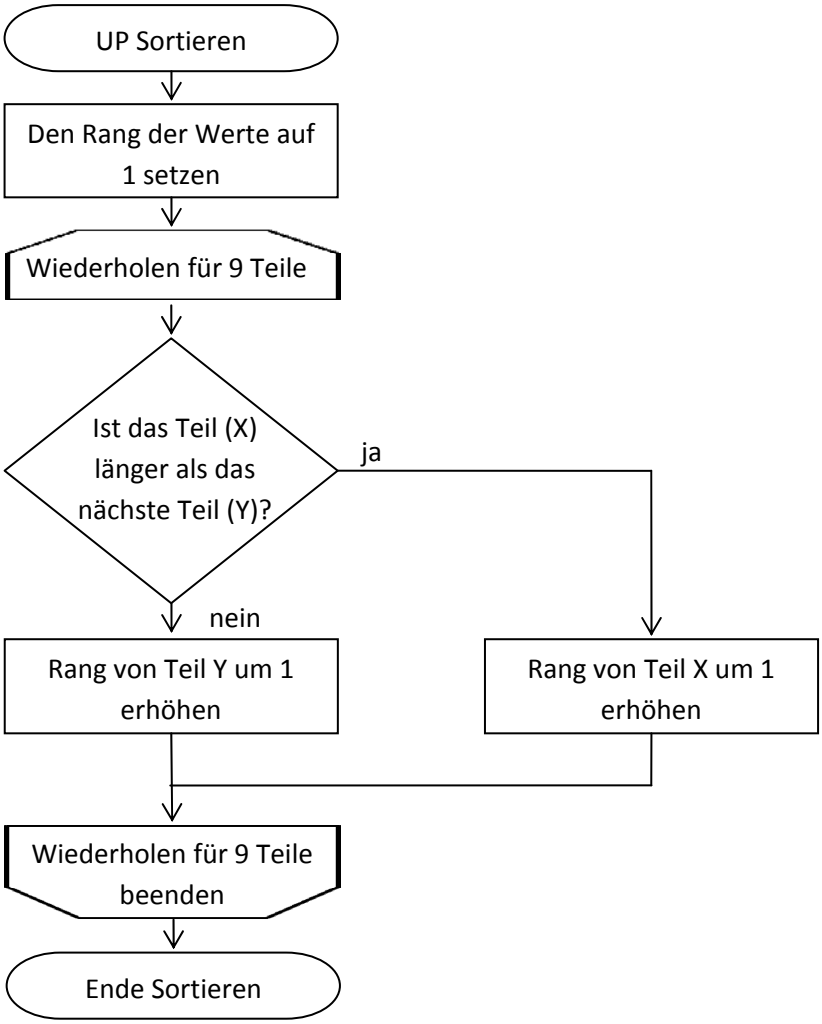


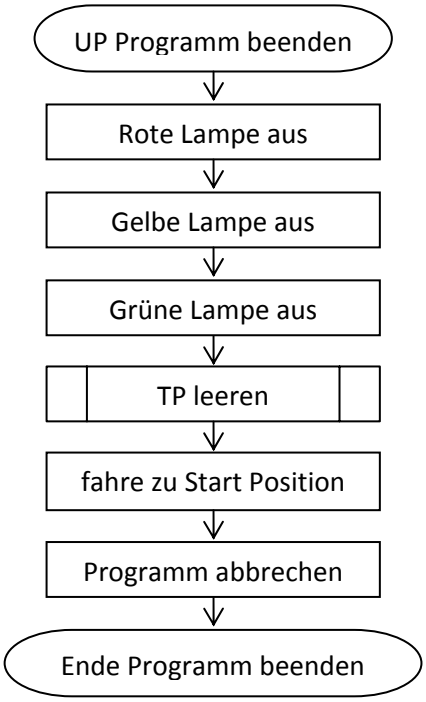

UP Bewegung(A,B)
A und B sind Position Parameter
Greifer öffnen
Re=false
fahre über A
fahre zu A
Greifer schließen
Fehlerkontrolle(1)
ja
Ist Re True?
nein
fahre über A
fahre über B
fahre zu B
Fehlerkontrolle(2)
Ende Bewegung


UP Fehlerkontrolle(F)
F ist für Fehlerart
kein Teil und F=1 ?
nein
ja
Gelbe Lampe an
Greifer öffnen
fahre 9cm nach oben
Error Message
Warte bis f1 oder f3 oder f5 gedrückt ist
kein Teil und F=2 ?
ja
Gelbe Lampe an
Greifer öffnen
Error Message
Warte bis f3 oder f5 gedrückt ist
f1 gedrückt ?
nein
ja
Try again
f3 gedrückt ?
nein
ja
Ignore
f5 gedrückt ?
ja
Cancel
Ende Fehlerkontrolle


P0
P2
P1


UP Messen
fahre zu P1
Wenn
Sensor aus
Bewegung abbrechen
Speicher die aktuelle
Position in P2
Länge = P0 – P2
fahre über P0
Ende Messen


UP Sortieren
Den Rang der Werte auf
1 setzen
Wiederholen für 9 Teile
Ist das Teil (X)
länger als das
nächste Teil (Y)?
ja
nein
Rang von Teil Y um 1
erhöhen
Rang von Teil X um 1
erhöhen
Wiederholen für 9 Teile
beenden
Ende Sortieren



(In dem oben abgebildetet Programmablaufplan wurden nur die wichtigsten Routinen erwähnt)

3.2 Karel-Programm

```
                                    Projektarbeit Industrieroboter Fanuc LR Mate 200i
--Gruppe :                          Wibke Lorenz, Mahmood Shubbak, Peng Xu
--techniche Daten Fanuc LR Mate 200i :  - 6 Kontroll-Achsen (J1, J2, J3, J4, J5, J6)
--                                  - R-J3-Steuerung
--Projektstart:                     WS 2010
--Revisionsnummer:                  00
--Dateiname:                        Projekt4.kl
```

```
PROGRAM Projekt4

VAR
S ,D                : ARRAY[9] OF XYZWPREXT   -- Entnahme- und Ablagepositionen
L                   : ARRAY[9] OF REAL        -- gemesssene Länge
V,Ind               : ARRAY[9] OF INTEGER     -- Rang-Array
i , j , n           : INTEGER                 -- Zählvariablen
sav_tcp,sav_npv     : POSITION                -- Werkzeug- und Anwenderkoordinaten
Home_Pos            : POSITION                -- Start Position
P0,P1,P2            : XYZWPREXT               -- Messpositionen
 CP,A1,B1,A2        : XYZWPREXT               -- Hilfspositionen
Re                  : BOOLEAN                 -- Reset-Merker
```

```
-- Rote Lampen an
ROUTINE Rot
        BEGIN
        DOUT[3]   = OFF     -- große grüne Meldeleuchte
        DOUT[12] = OFF      -- kleine grüne Meldeleuchte
        DOUT[2]   = OFF     -- große gelbe Meldeleuchte
        DOUT[11]= OFF       -- kleine gelbe Meldeleuchte
        DOUT[1]   = ON      -- große rote Meldeleuchte
        DOUT[10]= ON        -- kleine rote Meldeleuchte
END Rot
```

```
-- Gelbe Lampen an
ROUTINE Gelb
        BEGIN
        DOUT[3]   = OFF     -- große grüne Meldeleuchte
        DOUT[12] = OFF      -- kleine grüne Meldeleuchte
        DOUT[2]   = ON      -- große gelbe Meldeleuchte
        DOUT[11]= ON        -- kleine gelbe Meldeleuchte
        DOUT[1]   = OFF     -- große rote Meldeleuchte
        DOUT[10]= OFF       -- kleine rote Meldeleuchte
END Gelb
```

```
-- Grüne Lampen an
ROUTINE Grun
        BEGIN
        DOUT[3]   = ON      -- große grüne Meldeleuchte
        DOUT[12] = ON       -- kleine grüne Meldeleuchte
        DOUT[2]   = OFF     -- große gelbe Meldeleuchte
        DOUT[11]= OFF       -- kleine gelbe Meldeleuchte
        DOUT[1]   = OFF     -- große rote Meldeleuchte
        DOUT[10]= OFF       -- kleine rote Meldeleuchte
END Grun
```

```
 -- Bildschirm leeren
```

```
ROUTINE TP_leeren
        BEGIN
        WRITE TPDISPLAY(CHR(128))                    -- TP Display leeren
        WRITE TPFUNC(CHR(128))                       -- Funktionsleiste leeren
END TP_leeren
```

```
 -- Programm abbrechen
ROUTINE End_Pro
        BEGIN
        RDO[2]  = OFF                                -- Greifer öffnen
        DOUT[3]   = OFF                              -- Lampen aus
        DOUT[12]= OFF
        DOUT[2]  = OFF
        DOUT[11]= OFF
        DOUT[1]   = OFF
        DOUT[10]= OFF
        TP_leeren                                    -- Bildschirm leeren
        MOVE TO Home_Pos                             -- Bewegung zur Start Position
        ABORT
END End_Pro
```

```
 -- Fehler ignorieren
ROUTINE Ignore
        BEGIN
        Grun                                         -- Grüne Lampen an
        TP_leeren                                    -- Bildschirm leeren
        Re=FALSE                                     -- keine Wiederholung
END Ignore
```

```
 -- Vorgang wiederholen
ROUTINE Again
        BEGIN
        Grun                                         -- Grüne Lampen an
        TP_leeren                                    -- Bildschirm leeren
        Re=TRUE                                      -- wiederholen
END Again
```

```
-- Not-Aus Meldung
ROUTINE Not_Aus
        BEGIN
        Rot                                          -- Rote Lampen an
        TP_leeren                                    -- Bildschirm leeren
        force_spmenu(1,spi_tpuser,1)                 --Ausgabe auf dem TP
        SET_CURSOR(TPDISPLAY,1,7,0)
        WRITE ('Not-Aus wurde betätigt!!',cr,cr)
        WRITE ('Nach Fehlerbehebung Not-Aus lösen und',cr)
        WRITE ('ein Reset Taster drücken und starten mit Cycle Start!',cr,cr)
        WRITE ('Achtung Programm läuft weiter!!')
        WAIT FOR (OPIN[1]=ON) OR (TPIN[153]=ON)      -- Tasterkontrolle
        Grun                                         -- Grüne Lampe an
END Not_Aus
```

```
--  Fehler prüfen
ROUTINE Fehler(f:INTEGER)                            -- f=Fehlerart, 1: Teil nicht gegriffen
        BEGIN                                        --                2 : Teil verloren
```

```
        IF (RDI[2]=OFF) AND (f=1) THEN                          -- Fehlerkontrolle für f=1
            Gelb                                                -- Gelbe Lampen an
            RDO[2]=OFF                                          -- Greifer öffnen
            CP=CURPOS(0,0)
            CP.Z=CP.Z+90                                        -- 90mm nach oben fahren
            MOVE TO CP
            force_spmenu(1,spi_tpuser,1)                        -- Fehlerausgabe auf dem TP
            TP_leeren
            SET_CURSOR(TPDISPLAY,2,17,0)
            WRITE ('ERROR !!')
            SET_CURSOR(TPDISPLAY,4,7,0)
            WRITE ('Teil nicht vorhanden')
            WRITE TPFUNC ('Try again      Ignore        Cancel')
            WAIT FOR (TPIN[129]=ON) OR (TPIN[132]=ON) OR (TPIN[134]=ON)          -- Tasterkontrolle
            IF(TPIN[132]=ON) THEN                               -- F3 gedrückt
                Ignore                                          -- Fehler ignorieren
            ENDIF
            IF(TPIN[134]=ON) THEN                               -- F5 gedrückt
                End_Pro                                         -- Programm abbrechen
            ENDIF
            IF(TPIN[129]=ON) THEN                               -- F1 gedrückt
                Again                                           -- Vorgang wiederholen
            ENDIF
        ENDIF

        IF (RDI[2]=OFF) AND (f=2) THEN                          -- Fehlerkontrolle für f=2
            RDO[2]=OFF                                          -- Greifer öffnen
            Gelb                                                -- Gelbe Lampen an
            force_spmenu(1,spi_tpuser,1)                        -- Fehlerausgabe auf dem TP
            TP_leeren
            SET_CURSOR(TPDISPLAY,2,17,0)
            WRITE ('ERROR !!')
            SET_CURSOR(TPDISPLAY,4,7,0)
            WRITE ('Teil verloren')
            WRITE TPFUNC ('             Ignore        Cancel')
            WAIT FOR (TPIN[134]=ON) OR (TPIN[132]=ON)           -- Tasterkontrolle
            IF(TPIN[132]=ON) THEN                               -- F3 gedrückt
                Ignore                                          -- Fehler ignorieren
            ENDIF
            IF(TPIN[134]=ON) THEN                               -- F5 gedrückt
                End_Pro                                         -- Programm abbrechen
            ENDIF
        ENDIF
END Fehler
```

```
-- Index und Wert vom Rang-array V wechseln
ROUTINE Magic
        BEGIN
        FOR i = 1 TO 9 DO
            FOR j = 1 TO 9 DO
                IF V[i]=j THEN
                    Ind[j]=i
                ENDIF
            ENDFOR
        ENDFOR
```

```
END Magic
```

```
-- Transportieren der Teile
ROUTINE Mov(A,B:XYZWPREXT)
        BEGIN
        A1=A
        A1.Z=A1.Z+185
        A2=A
        A2.Z=A2.Z+25
        RDO[2]=OFF
Ag::    Re=FALSE
        MOVE TO A1                                      -- fahre 185mm über A
        WITH $SPEED=310,$MOTYPE=LINEAR MOVE TO A2       -- fahre  25mm über A
        WITH $SPEED=200,$MOTYPE=LINEAR MOVE TO A        -- fahre zu A
        RDO[2]=ON                                       -- Greifer schließen
        DELAY 300                                       -- 0,3s warten
        Fehler(1)
        IF Re THEN                                      -- Fehlerkontrolle 1
            GO TO Ag                                    -- Vorgang wiederholen
        ENDIF
        WITH $SPEED=1150,$MOTYPE=LINEAR MOVE TO A1      -- fahre 185mm über A
        B1=B
        IF n = 1 THEN                                   -- B ist Sensorposition P0
            B1.Z=B1.Z+85                                -- B1 ist 85mm über B
        ENDIF
        IF n = 2 THEN                                   -- B ist Ablageposition D
            B1.Z=B1.Z+155                               -- B1 ist 155mm über B
        ENDIF
        MOVE TO B1                                      -- fahre zu B1
        WITH $SPEED=300,$MOTYPE=LINEAR MOVE TO B        -- fahre zu B
        Fehler(2)                                       -- Fehlerkontrolle 2
END Mov
```

```
-- Längenmessung
ROUTINE Mess(Num:INTEGER)
        BEGIN
        WITH $SPEED=200,$MOTYPE=LINEAR,$TERMTYPE=FINE MOVE TO P1,        -- fahre zu Messposition P1
        WHEN DIN[10]+ DO                                -- Sensor aus
        CANCEL                                          -- Bewegung stoppen
        ENDMOVE
        P2=CURPOS(0,0)
        L[Num]=P0.Z-P2.Z                                -- Längenberechnung
        WITH $SPEED=950, $MOTYPE=LINEAR MOVE TO B1      -- fahre 85mm über P0
END Mess
```

```
-- Länge sortieren
ROUTINE Sort
        BEGIN
        FOR i = 1 TO 9 DO
            V[i]=1                                      -- Werte auf 1 setzen
        ENDFOR
        FOR i = 1 TO 8 DO
            FOR j = i+1 TO 9 DO
                IF L[i]>=L[j] THEN                      -- Sortieren der Länge
                    V[i]= V[i]+1
```

```
            ELSE
                V[j]= V[j]+1
            ENDIF
        ENDFOR
    ENDFOR
END Sort
```

```
-- Hauptprogramm
BEGIN

CONDITION[1]:                                          -- Not-Aus Kontrolle
    WHEN ERROR[11001] OR ERROR[11002] OR ERROR[11007] OR ERROR[11038] DO
        UNPAUSE
        Not_Aus                                        -- Not-Aus Meldung
        ENABLE CONDITION[1]
ENDCONDITION

    ENABLE CONDITION[1]

    $GROUP[1].$UTOOL=sav_tcp                           -- Werkzeugkoordinaten
    $GROUP[1].$UFRAME=sav_npv                          -- Anwenderkoordinaten
    $MOTYPE=JOINT
    $TERMTYPE=NODECEL
    $SPEED=2000

    Re=FALSE                                           -- Re initialisieren
    DOUT[9] = ON                                       -- weiße Lampe an
    WAIT FOR DIN[9]+                                   -- Start Taster
    DOUT[9] = OFF                                      -- weiße Lampe aus
    Grun                                               -- Grüne Lampen an
    MOVE TO Home_Pos                                   -- fahre zu Start Position
    FOR i= 1 TO 9 DO
        n=1
        Mov(S[i],P0)                                   -- 9x Teil zum Sensor transportieren
        Mess(i)                                        -- Messen
        A1=S[i]
        A1.Z=A1.Z+180
        MOVE TO A1                                     -- 180mm über S Positionen fahren
        A2=S[i]
        A2.Z=A2.Z+25
        WITH $SPEED=310, $MOTYPE=LINEAR MOVE TO A2     -- 25mm über S Positionen fahren
        WITH $SPEED=200, $MOTYPE=LINEAR,$TERMTYPE=FINE MOVE TO S[i]     -- fahre zu S Positionen
        RDO[2]=OFF                                     -- Greifer öffnen
        DELAY 100                                      -- Wartezeit
        CP=CURPOS(0,0)
        CP.Z=CP.Z+50
        MOVE TO CP                                     -- 50mm nach oben fahren
    ENDFOR
    Sort                                               -- Länge sortieren
    Magic                                              -- Index und Wert vom Rang-array V wechseln
    FOR i= 9 DOWNTO 1 DO                               -- Teile sortieren
        n=2
        Mov(S[Ind[i]],D[i])
        RDO[2]=OFF                                     -- Greifer öffnen
        DELAY 100                                      -- Wartezeit
        CP=CURPOS(0,0)
```

```
            CP.Z=CP.Z+50                                    -- 50mm nach oben fahren
            MOVE TO CP
        ENDFOR
        End_Pro                                             -- Programm abbrechen
END Projekt4
```