

8.2.1 Industrielle Handhabungssysteme "Externe Signale und Gerätekommunikation"

Ziele

Die Ziele der Übung "Externe Signale und Gerätekommunikation" umfassen das Verständnis des IO-Boards sowie den Austausch von Daten zwischen dem Kuka-Roboter und dem Schaltboard. Der Roboter soll in der Lage sein, Eingänge auszulesen und Ausgänge anzusteuern. Die Steuerung soll abhängig von den Eingängen und der Position des Roboter-Greifarms die Ausgänge schalten, um Bauteile vom Fließband zu bewegen.

Wir verfügen über Inputs und Outputs:

- Wir steuern die Ventile für die Druckluft, um das Fließband zu bewegen.
- Zusätzlich gibt es zwei Tasten, eine Wahlschaltung und eine Signallampe.
- Die Zylinder sind mit Positionssensoren ausgestattet, die rückmelden, ob sie ausgefahren oder nicht ausgefahren sind.
- Der Motor des Förderbands ist ebenfalls ein Output (zusätzlich abgesichert mit einem Not-Halt nur für den Motor, wiederanlaufgeschützt).

Es gibt insgesamt 8 Bauteile an 8 unterschiedlichen Positionen. Das Programm soll folgende Funktionen umfassen:

Herausforderungen

1. Sinnvollen Ablauf überlegen.
2. Applikation selbst gestalten (Dialogfeld).
3. Eingaben (JOptionPane), die vom Nutzer kommen, verarbeiten (I/Os ansprechen).
4. "Das gemeine ä"

Tätigkeiten der Gruppe

- Werkzeug am Kuka Roboter anbringen: Der Sauger am Adapter des Werkzeugwechslers am Kuka Roboter musste angebracht werden.

Lösungsansätze:

- Flussdiagramm erstellen.
- Code schreiben.
- Testen.

Flussdiagramm

1. Acht Bauteile liegen auf der Palette (die Palette befindet sich wartend in der Ausgangsposition).
2. Benutzer wählt im Interface (Eingabefeld/Dialogfeld über die Steuerkonsole) ein Bauteil anhand der ID (=Position) aus.
3. Verhindern, dass eine neue Bestellung aufgegeben wird.

4. (Optional) Manuellmodus - manueller Override (Förderband fahren/stoppen mittels Schalter).
5. Roboter sendet eine Anforderung "Ich brauche eine Palette".
6. Förderband einschalten.
7. Palette fährt über das Fließband auf die Position.
8. Endschalter sind relevant.
9. Palette spannen.
10. Palette sendet Bestätigung an den Roboter "Palette auf Position" (Meldung für den Benutzer).
11. Mit dem Greifarm über die Palette fahren und positionieren.
12. Bucht öffnen.
13. Entnehmen.
14. Roboter greift Bauteil von der gewünschten Position.
15. Roboter fährt zur Box.
16. Roboter lässt Bauteil los und wirft es somit in eine Box.
17. Meldung "Bauteil abgeliefert".
18. Eingabefeld für Bestellung freigeben (Lampe leuchtet grün).

Kommentierter Programmcode

Code in Kuka Robot Language (KRL) wird noch hinzugefügt.

Aufgetretene Probleme & daraus resultierende Erkenntnisse

1. Sauger funktioniert nicht weil falscher Schlauch angesteckt. Aktivierung nicht über Taste sondern über I/O.
- 2.

Fotos

Fragestellungen

1. Was versteht man unter "normally open" und "normally closed", wenn man von einem Taster spricht?
 - Normally Open (NO) und Normally Closed (NC) beschreiben den Standardzustand eines Tasters oder Schalters. Ein NO-Taster schließt den Stromkreis, wenn er gedrückt wird, während ein NC-Taster den Stromkreis im Ruhezustand schließt und beim Drücken öffnet.
2. Wie kann man Taster mit Eingangsmodulen verbinden?
 - Taster können direkt mit den Eingangsmodulen des IO-Boards verbunden werden, indem man die entsprechenden NO- oder NC-Kontakte nutzt.
3. Wie kann man LEDs oder Lampen mit Ausgangsmodulen verbinden?
 - Um LEDs mit den Ausgangsmodulen zu verbinden, können die Eingänge der Ausgangsmodule mit den entsprechenden Kontakten der LEDs oder Lampen verbunden werden.
4. Mit welchen Programmkonstrukten kann ein Greifer geöffnet und geschlossen werden?
 - Das Öffnen und Schließen eines Greifers kann über digitale Ausgangssignale gesteuert werden.

5. Wie können Ausgänge gesetzt werden?

- Ausgänge können mit den Befehlen **OUT** oder **SET** gesetzt werden.

6. Wie können Eingänge ausgelesen werden?

- Eingänge werden mit dem Befehl **IN** ausgelesen, um den Zustand von Sensoren oder Tastern zu überprüfen.

7. Mit welchem Programmkonstrukt kann auf das Setzen eines Einganges gewartet werden?

- Um auf ein spezifisches Eingangssignal zu warten, kann das Konstrukt **WAIT FOR** verwendet werden. Dies ermöglicht dem Programm, die Ausführung zu pausieren, bis ein bestimmter Eingangszustand erkannt wird.

Code

Input Prompt

Aufgekommene Probleme:

- Parameter müssen 3 und 7 sein
- es werden keine Umlaute unterstützt

```
&ACCESS RVP
&REL 28
&PARAM DISKPATH = KRC:\R1\Program\sj202324
DECL GLOBAL BOOL myFlag = false
DEF Befoerderung ( )
DECL INT Bucht, zaehler
;DECL bool Eingabe
DECL KrlMsg_T Message1, Message2
DECL KrlMsgPar_T Parameter[3], Parameter2[3]
DECL KrlMsgOpt_T Opt, Opt2
DECL KrlMsgDlgSK_T Key[7], Key2[7]
DECL INT Ticket, Answer, Ticket2, Answer2
;FOLD INI;%{PE}
;FOLD BASISTECH INI
  GLOBAL INTERRUPT DECL 3 WHEN $STOPMESS==TRUE DO IR_STOPM ( )
  INTERRUPT ON 3
  BAS (#INITMOV,0 )
;ENDFOLD (BASISTECH INI)
GLOBAL INTERRUPT DECL 4 WHEN $IN[32]==TRUE DO haltStopp( )
GLOBAL INTERRUPT DECL 5 WHEN $IN[33]==TRUE DO weitermachen()
GLOBAL INTERRUPT DECL 6 WHEN $IN[34]==TRUE DO rechtsfahren()
GLOBAL INTERRUPT DECL 7 WHEN $IN[35]==TRUE DO linksfahren()
;FOLD USER INI
  ;Make your modifications here
;ENDFOLD (USER INI)
;ENDFOLD (INI)
Message1 = {MODUL[] "USER", Nr 1508, MSG_TXT[] "Teil auswaehlen"}
Key[1]= {SK_type #value, SK_txt[] "Teil 1"}
```

```

Key[2]= {SK_type #value, SK_txt[] "Teil 2"}
Key[3]= {SK_type #value, SK_txt[] "Teil 3"}
Key[4]= {SK_type #value, SK_txt[] "Teil 4"}
Key[5]= {SK_type #value, SK_txt[] "Teil 5-8"}
;Key[6]= {SK_type #value, SK_txt[] "Teil 6"}
;Key[7]= {SK_type #value, SK_txt[] "Teil 7"}
;Key[8]= {SK_type #value, SK_txt[] "Teil 8"}
;Key[9]= {SK_type #empty}
Opt = {VL_Stop TRUE, Clear_P_Reset TRUE, Clear_P_SAW TRUE, Log_To_DB
FALSE}
Ticket = Set_Krldlg(Message1, Parameter[],Key[], Opt)
IF (Ticket > 0) THEN
    WHILE (Exists_Krldlg (Ticket, Answer))
        WAIT SEC 0
    ENDWHILE
    Bucht=29+Answer
ENDIF
IF (Answer == 5) THEN
    Message2 = {MODUL[] "USER", Nr 1508, MSG_TXT[] "Teil auswaehlen"}
    Key2[1]= {SK_type #value, SK_txt[] "Teil 5"}
    Key2[2]= {SK_type #value, SK_txt[] "Teil 6"}
    Key2[3]= {SK_type #value, SK_txt[] "Teil 7"}
    Key2[4]= {SK_type #value, SK_txt[] "Teil 8"}
    ;Opt = {VL_Stop TRUE, Clear_P_Reset TRUE, Clear_P_SAW TRUE, Log_To_DB
FALSE}
    Ticket2 = Set_Krldlg(Message2, Parameter2[],Key2[], Opt)
    IF (Ticket2 > 0) THEN
        WHILE (Exists_Krldlg (Ticket2, Answer))
            WAIT SEC 0
        ENDWHILE
        Answer = Answer +4
        Bucht=29+Answer
    ENDIF
ENDIF
$out[38]=false
$out[39]=false
$out[43]=true
$out[42]=false
zaehler = 0
WHILE $in[30]==false
    $out[42]=true
    IF( zaehler == 1)THEN
        $out[44]=true
        zaehler = 0
    ELSE
        $out[44]=false
        zaehler = 1
    ENDIF
ENDWHILE

    $out[44]=false
    $out[42]=false
    $out[38]=true
    $out[39]=true

```

```

$out[Bucht]=true
;FOLD SPTP P0 Vel=100 % PDAT1 Tool[1]:Kurtigreifer Base[0] ;%{PE}
;FOLD Parameters ;%{h}
;Params IlfProvider=kukaroboter.basistech.inlineforms.movement.spline;
Kuka.IsGlobalPoint=False; Kuka.PointName=P0; Kuka.BlendingEnabled=False;
Kuka.MoveDataPtpName=PDAT1; Kuka.VelocityPtp=100;
Kuka.VelocityFieldEnabled=True; Kuka.ColDetectFieldEnabled=True;
Kuka.CurrentCDSetIndex=0; Kuka.MovementParameterFieldEnabled=True;
IlfCommand=SPTP
;ENDFOLD
SPTP XP0 WITH $VEL_AXIS[1] = SVEL_JOINT(100.0), $TOOL = ST00L2(FP0), $BASE
=
SBASE(FP0.BASE_NO), $IPO_MODE = SIPO_MODE(FP0.IPO_FRAME), $LOAD =
SLOAD(FP0.TOOL_NO), $ACC_AXIS[1] = SACC_JOINT(PPDAT1), $APO =
SAPO_PTP(PPDAT1), $GEAR_JERK[1] = SGEAR_JERK(PPDAT1), $COLLMON_TOL_PRO[1]
=
USE_CM_PRO_VALUES(0)
;ENDFOLD
SWITCH Answer
CASE 1
;FOLD SLIN P1 Vel=2 m/s CPDAT1 Tool[1]:Kurtigreifer Base[0] ;%{PE}
;FOLD Parameters ;%{h}
;Params IlfProvider=kukaroboter.basistech.inlineforms.movement.spline;
Kuka.IsGlobalPoint=False; Kuka.PointName=P1; Kuka.BlendingEnabled=False;
Kuka.MoveDataName=CPDAT1; Kuka.VelocityPath=2;
Kuka.VelocityFieldEnabled=True; Kuka.ColDetectFieldEnabled=True;
Kuka.CurrentCDSetIndex=0; Kuka.MovementParameterFieldEnabled=True;
IlfCommand=SLIN
;ENDFOLD
SLIN XP1 WITH $VEL = SVEL_CP(2.0, , LCPDAT1), $TOOL = ST00L2(FP1), $BASE =
SBASE(FP1.BASE_NO), $IPO_MODE = SIPO_MODE(FP1.IPO_FRAME), $LOAD =
SLOAD(FP1.TOOL_NO), $ACC = SACC_CP(LCPDAT1), $ORI_TYPE =
SORI_TYP(LCPDAT1),
$APO = SAPO(LCPDAT1), $JERK = SJERK(LCPDAT1), $COLLMON_TOL_PRO[1] =
USE_CM_PRO_VALUES(0)
;ENDFOLD
CASE 2
;FOLD SLIN P2 Vel=2 m/s CPDAT2 Tool[1]:Kurtigreifer Base[0] ;%{PE}
;FOLD Parameters ;%{h}

;Params IlfProvider=kukaroboter.basistech.inlineforms.movement.spline;
Kuka.IsGlobalPoint=False; Kuka.PointName=P2; Kuka.BlendingEnabled=False;
Kuka.MoveDataName=CPDAT2; Kuka.VelocityPath=2;
Kuka.VelocityFieldEnabled=True; Kuka.ColDetectFieldEnabled=True;
Kuka.CurrentCDSetIndex=0; Kuka.MovementParameterFieldEnabled=True;
IlfCommand=SLIN
;ENDFOLD
SLIN XP2 WITH $VEL = SVEL_CP(2.0, , LCPDAT2), $TOOL = ST00L2(FP2), $BASE =
SBASE(FP2.BASE_NO), $IPO_MODE = SIPO_MODE(FP2.IPO_FRAME), $LOAD =
SLOAD(FP2.TOOL_NO), $ACC = SACC_CP(LCPDAT2), $ORI_TYPE =
SORI_TYP(LCPDAT2),
$APO = SAPO(LCPDAT2), $JERK = SJERK(LCPDAT2), $COLLMON_TOL_PRO[1] =
USE_CM_PRO_VALUES(0)
;ENDFOLD

```

```

CASE 3
;FOLD SLIN P3 Vel=2 m/s CPDAT3 Tool[1]:Kurtigreifer Base[0] ;%{PE}
;FOLD Parameters ;%{h}
;Params IlfProvider=kukaroboter.basistech.inlineforms.movement.spline;
Kuka.IsGlobalPoint=False; Kuka.PointName=P3; Kuka.BlendingEnabled=False;
Kuka.MoveDataName=CPDAT3; Kuka.VelocityPath=2;
Kuka.VelocityFieldEnabled=True; Kuka.ColDetectFieldEnabled=True;
Kuka.CurrentCDSetIndex=0; Kuka.MovementParameterFieldEnabled=True;
IlfCommand=SLIN
;ENDFOLD
SLIN XP3 WITH $VEL = SVEL_CP(2.0, , LCPDAT3), $TOOL = ST00L2(FP3), $BASE =
SBASE(FP3.BASE_NO), $IPO_MODE = SIPO_MODE(FP3.IPO_FRAME), $LOAD =
SLOAD(FP3.TOOL_NO), $ACC = SACC_CP(LCPDAT3), $ORI_TYPE =
SORI_TYP(LCPDAT3),
$APO = SAPO(LCPDAT3), $JERK = SJERK(LCPDAT3), $COLLMON_TOL_PRO[1] =
USE_CM_PRO_VALUES(0)
;ENDFOLD
CASE 4
;FOLD SLIN P4H Vel=2 m/s CPDAT11 Tool[1]:Kurtigreifer Base[0] ;%{PE}
;FOLD Parameters ;%{h}
;Params IlfProvider=kukaroboter.basistech.inlineforms.movement.spline;
Kuka.IsGlobalPoint=False; Kuka.PointName=P4H; Kuka.BlendingEnabled=False;
Kuka.MoveDataName=CPDAT11; Kuka.VelocityPath=2;
Kuka.VelocityFieldEnabled=True; Kuka.ColDetectFieldEnabled=True;
Kuka.CurrentCDSetIndex=0; Kuka.MovementParameterFieldEnabled=True;
IlfCommand=SLIN
;ENDFOLD
SLIN XP4H WITH $VEL = SVEL_CP(2.0, , LCPDAT11), $TOOL = ST00L2(FP4H),
$BASE
= SBASE(FP4H.BASE_NO), $IPO_MODE = SIPO_MODE(FP4H.IPO_FRAME), $LOAD =
SLOAD(FP4H.TOOL_NO), $ACC = SACC_CP(LCPDAT11), $ORI_TYPE =
SORI_TYP(LCPDAT11), $APO = SAPO(LCPDAT11), $JERK = SJERK(LCPDAT11),
$COLLMON_TOL_PRO[1] = USE_CM_PRO_VALUES(0)

;ENDFOLD
;FOLD SLIN P4 Vel=2 m/s CPDAT9 Tool[1]:Kurtigreifer Base[0] ;%{PE}
;FOLD Parameters ;%{h}
;Params IlfProvider=kukaroboter.basistech.inlineforms.movement.spline;
Kuka.IsGlobalPoint=False; Kuka.PointName=P4; Kuka.BlendingEnabled=False;
Kuka.MoveDataName=CPDAT9; Kuka.VelocityPath=2;
Kuka.VelocityFieldEnabled=True; Kuka.ColDetectFieldEnabled=True;
Kuka.CurrentCDSetIndex=0; Kuka.MovementParameterFieldEnabled=True;
IlfCommand=SLIN
;ENDFOLD
SLIN XP4 WITH $VEL = SVEL_CP(2.0, , LCPDAT9), $TOOL = ST00L2(FP4), $BASE =
SBASE(FP4.BASE_NO), $IPO_MODE = SIPO_MODE(FP4.IPO_FRAME), $LOAD =
SLOAD(FP4.TOOL_NO), $ACC = SACC_CP(LCPDAT9), $ORI_TYPE =
SORI_TYP(LCPDAT9),
$APO = SAPO(LCPDAT9), $JERK = SJERK(LCPDAT9), $COLLMON_TOL_PRO[1] =
USE_CM_PRO_VALUES(0)
;ENDFOLD
;FOLD Gripper SET [2]Sauger 1 State=[1]ON Wait 2[s] Check with Strategy 1
;%
{PE}

```

```

;FOLD Parameters Parameters ;%{h}
;Params
IfProvider=GripperTech.GripperSet;setgripper=2;setstate=1;setcont=nocont;
wa
ittime=2;errorstrategy=1
;ENDFOLD
GRPg_SetStateAndCheck(2, 1, 2, 1)
;ENDFOLD
;FOLD SLIN P4H Vel=2 m/s CPDAT10 Tool[1]:Kurtigreifer Base[0] ;%{PE}
;FOLD Parameters ;%{h}
;Params IfProvider=kukaroboter.basistech.inlineforms.movement.spline;
Kuka.IsGlobalPoint=False; Kuka.PointName=P4H; Kuka.BlendingEnabled=False;
Kuka.MoveDataName=CPDAT10; Kuka.VelocityPath=2;
Kuka.VelocityFieldEnabled=True; Kuka.ColDetectFieldEnabled=True;
Kuka.CurrentCDSetIndex=0; Kuka.MovementParameterFieldEnabled=True;
IfCommand=SLIN
;ENDFOLD
SLIN XP4H WITH $VEL = SVEL_CP(2.0, , LCPDAT10), $TOOL = ST00L2(FP4H),
$BASE
= SBASE(FP4H.BASE_NO), $IPO_MODE = SIPO_MODE(FP4H.IPO_FRAME), $LOAD =
SLOAD(FP4H.TOOL_NO), $ACC = SACC_CP(LCPDAT10), $ORI_TYPE =
SORI_TYP(LCPDAT10), $APO = SAPO(LCPDAT10), $JERK = SJERK(LCPDAT10),
$COLLMON_TOL_PRO[1] = USE_CM_PRO_VALUES(0)
;ENDFOLD
CASE 5
;FOLD SLIN P5 Vel=2 m/s CPDAT4 Tool[1]:Kurtigreifer Base[0] ;%{PE}
;FOLD Parameters ;%{h}
;Params IfProvider=kukaroboter.basistech.inlineforms.movement.spline;
Kuka.IsGlobalPoint=False; Kuka.PointName=P5; Kuka.BlendingEnabled=False;

Kuka.MoveDataName=CPDAT4; Kuka.VelocityPath=2;
Kuka.VelocityFieldEnabled=True; Kuka.ColDetectFieldEnabled=True;
Kuka.CurrentCDSetIndex=0; Kuka.MovementParameterFieldEnabled=True;
IfCommand=SLIN
;ENDFOLD
SLIN XP5 WITH $VEL = SVEL_CP(2.0, , LCPDAT4), $TOOL = ST00L2(FP5), $BASE =
SBASE(FP5.BASE_NO), $IPO_MODE = SIPO_MODE(FP5.IPO_FRAME), $LOAD =
SLOAD(FP5.TOOL_NO), $ACC = SACC_CP(LCPDAT4), $ORI_TYPE =
SORI_TYP(LCPDAT4),
$APO = SAPO(LCPDAT4), $JERK = SJERK(LCPDAT4), $COLLMON_TOL_PRO[1] =
USE_CM_PRO_VALUES(0)
;ENDFOLD
CASE 6
;FOLD SLIN P6 Vel=2 m/s CPDAT5 Tool[1]:Kurtigreifer Base[0] ;%{PE}
;FOLD Parameters ;%{h}
;Params IfProvider=kukaroboter.basistech.inlineforms.movement.spline;
Kuka.IsGlobalPoint=False; Kuka.PointName=P6; Kuka.BlendingEnabled=False;
Kuka.MoveDataName=CPDAT5; Kuka.VelocityPath=2;
Kuka.VelocityFieldEnabled=True; Kuka.ColDetectFieldEnabled=True;
Kuka.CurrentCDSetIndex=0; Kuka.MovementParameterFieldEnabled=True;
IfCommand=SLIN
;ENDFOLD
SLIN XP6 WITH $VEL = SVEL_CP(2.0, , LCPDAT5), $TOOL = ST00L2(FP6), $BASE =
SBASE(FP6.BASE_NO), $IPO_MODE = SIPO_MODE(FP6.IPO_FRAME), $LOAD =

```

```

SLOAD(FP6.TOOL_N0), $ACC = SACC_CP(LCPDAT5), $ORI_TYPE =
SORI_TYP(LCPDAT5),
$APO = SAPO(LCPDAT5), $JERK = SJERK(LCPDAT5), $COLLMON_TOL_PRO[1] =
USE_CM_PRO_VALUES(0)
;ENDFOLD
CASE 7
;FOLD SLIN P7 Vel=2 m/s CPDAT6 Tool[1]:Kurtigreifer Base[0] ;%{PE}
;FOLD Parameters ;%{h}
;Params IlfProvider=kukaroboter.basistech.inlineforms.movement.spline;
Kuka.IsGlobalPoint=False; Kuka.PointName=P7; Kuka.BlendingEnabled=False;
Kuka.MoveDataName=CPDAT6; Kuka.VelocityPath=2;
Kuka.VelocityFieldEnabled=True; Kuka.ColDetectFieldEnabled=True;
Kuka.CurrentCDSetIndex=0; Kuka.MovementParameterFieldEnabled=True;
IlfCommand=SLIN
;ENDFOLD
SLIN XP7 WITH $VEL = SVEL_CP(2.0, , LCPDAT6), $TOOL = STOOL2(FP7), $BASE =
SBASE(FP7.BASE_N0), $IPO_MODE = SIPO_MODE(FP7.IPO_FRAME), $LOAD =
SLOAD(FP7.TOOL_N0), $ACC = SACC_CP(LCPDAT6), $ORI_TYPE =
SORI_TYP(LCPDAT6),
$APO = SAPO(LCPDAT6), $JERK = SJERK(LCPDAT6), $COLLMON_TOL_PRO[1] =
USE_CM_PRO_VALUES(0)
;ENDFOLD
CASE 8

;FOLD SLIN P8 Vel=2 m/s CPDAT7 Tool[1]:Kurtigreifer Base[0] ;%{PE}
;FOLD Parameters ;%{h}
;Params IlfProvider=kukaroboter.basistech.inlineforms.movement.spline;
Kuka.IsGlobalPoint=False; Kuka.PointName=P8; Kuka.BlendingEnabled=False;
Kuka.MoveDataName=CPDAT7; Kuka.VelocityPath=2;
Kuka.VelocityFieldEnabled=True; Kuka.ColDetectFieldEnabled=True;
Kuka.CurrentCDSetIndex=0; Kuka.MovementParameterFieldEnabled=True;
IlfCommand=SLIN
;ENDFOLD
SLIN XP8 WITH $VEL = SVEL_CP(2.0, , LCPDAT7), $TOOL = STOOL2(FP8), $BASE =
SBASE(FP8.BASE_N0), $IPO_MODE = SIPO_MODE(FP8.IPO_FRAME), $LOAD =
SLOAD(FP8.TOOL_N0), $ACC = SACC_CP(LCPDAT7), $ORI_TYPE =
SORI_TYP(LCPDAT7),
$APO = SAPO(LCPDAT7), $JERK = SJERK(LCPDAT7), $COLLMON_TOL_PRO[1] =
USE_CM_PRO_VALUES(0)
;ENDFOLD
ENDSWITCH
;FOLD Gripper SET [3]Sauger 2 State=[1]ON Wait 0.5[s] Check with Strategy
1
;%{PE}
;FOLD Parameters Parameters ;%{h}
;Params
IlfProvider=GripperTech.GripperSet;setgripper=3;setstate=1;setcont=nocont;
wa
ittime=0.5;errorstrategy=1
;ENDFOLD
GRPg_SetStateAndCheck(3, 1, 0.5, 1)
;ENDFOLD
;FOLD SLIN P0 Vel=2 m/s CPDAT8 Tool[1]:Kurtigreifer Base[0] ;%{PE}
;FOLD Parameters ;%{h}

```



```

;Params IlfProvider=kukaroboter.basistech.inlineforms.movement.spline;
Kuka.IsGlobalPoint=False; Kuka.PointName=P0; Kuka.BlendingEnabled=False;
Kuka.MoveDataName=CPDAT8; Kuka.VelocityPath=2;
Kuka.VelocityFieldEnabled=True; Kuka.ColDetectFieldEnabled=True;
Kuka.CurrentCDSetIndex=0; Kuka.MovementParameterFieldEnabled=True;
IlfCommand=SLIN
;ENDFOLD
SLIN XP0 WITH $VEL = SVEL_CP(2.0, , LCPDAT8), $TOOL = ST00L2(FP0), $BASE =
SBASE(FP0.BASE_NO), $IPO_MODE = SIPO_MODE(FP0.IPO_FRAME), $LOAD =
SLOAD(FP0.TOOL_NO), $ACC = SACC_CP(LCPDAT8), $ORI_TYPE =
SORI_TYP(LCPDAT8),
$APO = SAPO(LCPDAT8), $JERK = SJERK(LCPDAT8), $COLLMON_TOL_PRO[1] =
USE_CM_PRO_VALUES(0)
;ENDFOLD
;FOLD SPTP PFIN Vel=100 % PDAT2 Tool[1]:Kurtigreifer Base[0]   ;%{PE}
;FOLD Parameters ;%{h}
;Params IlfProvider=kukaroboter.basistech.inlineforms.movement.spline;

    Kuka.IsGlobalPoint=False; Kuka.PointName=PFIN;
Kuka.BlendingEnabled=False;
Kuka.MoveDataPtpName=PDAT2; Kuka.VelocityPtp=100;
Kuka.VelocityFieldEnabled=True; Kuka.ColDetectFieldEnabled=True;
Kuka.CurrentCDSetIndex=0; Kuka.MovementParameterFieldEnabled=True;
IlfCommand=SPTP
;ENDFOLD
SPTP XPFIN WITH $VEL_AXIS[1] = SVEL_JOINT(100.0), $TOOL = ST00L2(FPFIN),
$BASE = SBASE(FPFIN.BASE_NO), $IPO_MODE = SIPO_MODE(FPFIN.IPO_FRAME),
$LOAD
= SLOAD(FPFIN.TOOL_NO), $ACC_AXIS[1] = SACC_JOINT(PPDAT2), $APO =
SAPO_PTP(PPDAT2), $GEAR_JERK[1] = SGEAR_JERK(PPDAT2), $COLLMON_TOL_PRO[1]
=
USE_CM_PRO_VALUES(0)
;ENDFOLD
;FOLD Gripper SET [3]Sauger 2 State=[2]OFF Wait 0.2[s] Check with Strategy
1
;{%{PE}}
;FOLD Parameters Parameters ;%{h}
;Params
IlfProvider=GripperTech.GripperSet;setgrripper=3;setstate=2;setcont=nocont;
wa
ittime=0.2;errorstrategy=1
;ENDFOLD
GRPg_SetStateAndCheck(3, 2, 0.2, 1)
;ENDFOLD
;FOLD Gripper SET [2]Sauger 1 State=[2]OFF Wait 0.5[s] Check with Strategy
1
;{%{PE}}
;FOLD Parameters Parameters ;%{h}
;Params
IlfProvider=GripperTech.GripperSet;setgrripper=2;setstate=2;setcont=nocont;
wa
ittime=0.5;errorstrategy=1
;ENDFOLD
GRPg_SetStateAndCheck(2, 2, 0.5, 1)

```

```
;ENDFOLD
$out[38]=false
$out[39]=false
$out[43]=true
$out[42]=false
$out[Bucht]=false
END
DEF haltStopp()
  INTERRUPT ON 4
  BRAKE

END
DEF weitermachen()
  INTERRUPT ON 5
  RESUME

END
DEF rechtsfahren()
  INTERRUPT ON 5
  IF(myFlag == TRUE) THEN
    $out[38]=false
    $out[39]=false
    $out[43]=true
    WHILE $in[30]==false THEN
      $out[42]=true
    ENDWHILE
  ENDIF
  RESUME END
DEF linksfahren()
  INTERRUPT ON 5
  IF(myFlag == TRUE) THEN
    $out[38]=false
    $out[39]=false
    $out[43]=false
    WHILE $in[31]==false THEN
      $out[42]=true
    ENDWHILE
  ENDIF
  RESUME END
```