

List Protokoll

Team: Kacper Bohaczyk, Kevin Bauer, Lionel Ruf, Leopold Kernegger, Marko Djordjevic, Luan Sheriffi, Drinor Sutaj, Francesco Polichetti, Danko Vukoja

Date: 30-April-2024

Aufgabe: Industrielle Handhabungssysteme "Externe Signale und Gerätekommunikation"

Voraussetzungen

Roboterarm

Laptop

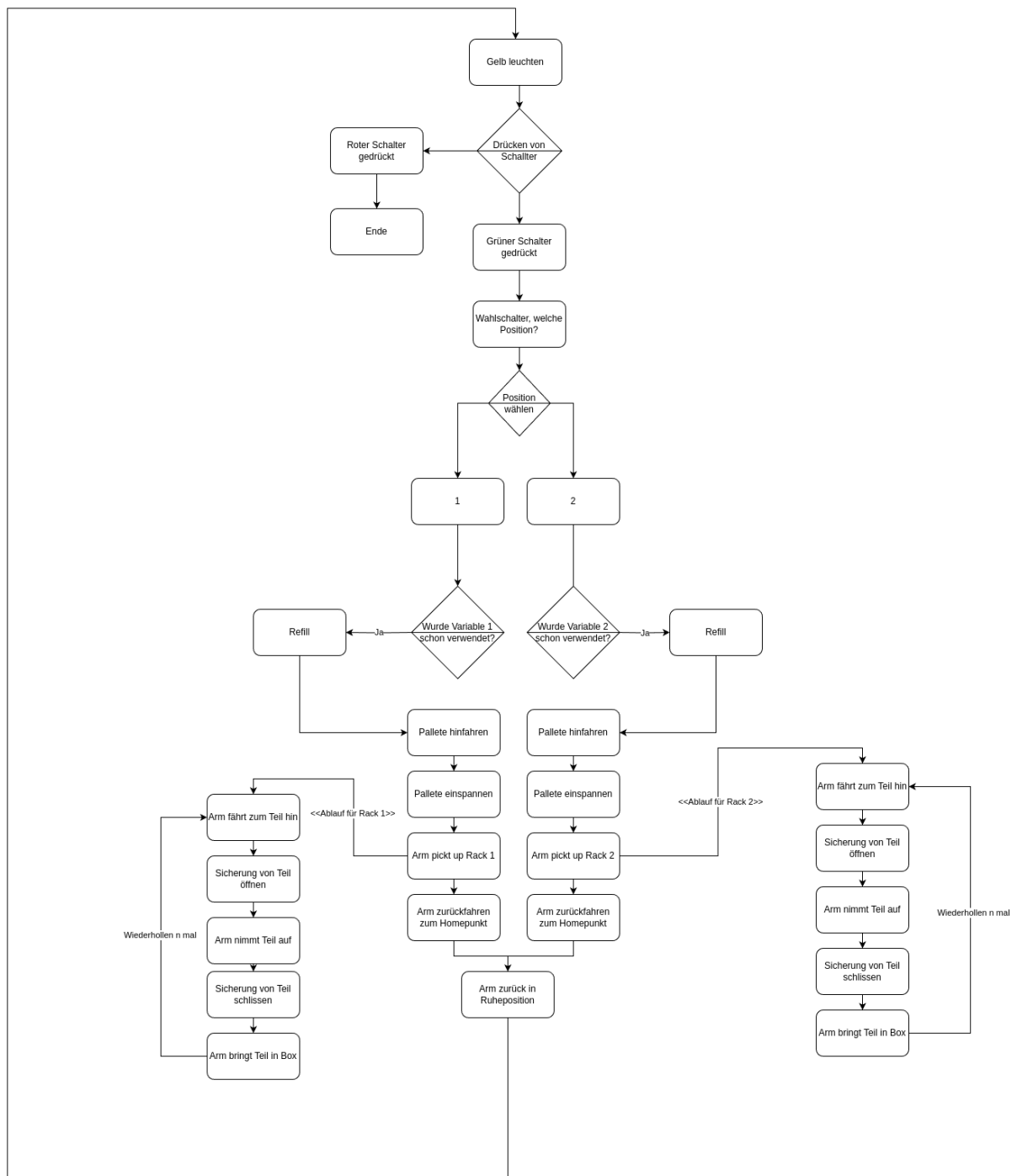
Programmierungsumgebung

Wissen

Guide

1. Ablauf planen
2. Ablauf-UML Diagramm erstellen
3. Punkte für Roboterarm programmieren
4. Bewegungen des Roboterarms programmieren
5. Code schreiben Cobol
6. Recherche

UML-Ablaufdiagramm



Roboterdiagramm Programmieren

1. Befehlspanel nehmen
2. Zum Punkten hinfahren
3. Punkte speichern
4. Logik implementieren

Sonstige Überlegungen

Pallette in zwei Racks geteilt(der Länge nach), da die Beschriftung der einzelnen Halterungen für die Teile bereits so war

Programmierung der Buttons sowie Leuchten:

- Schieberegler nach Links (Modus 1)
- Schieberegler nach Rechts (Modus 2)

Roter Knopf --> Programm beenden

Grüner Knopf --> Auswahl bestätigen / Modus starten

Gelbe Lampe:

- Leuchtend --> Bereit für Eingabe
- Blinkend --> Rack leer muss aufgefüllt werden

Roboterarm bewegen:

- Von Homepunkt zu leicht über den Teilen: PTP
- Von leicht über den Teilen zum Teil: linear

Genauso beim hinterlassen in der Box

--> Hilfreich für Ansteuerung des Verschlussmechanismus der Teile, denn:

- Halterung wird geöffnet wenn Roboterarm über dem Teil angelangt ist (Vor linearer Bewegung runter zum Zielobjekt)
- Halterung wird geschlossen wenn Roboterarm wieder über der Ursprünglichen Position des Teil angelangt ist (nach linearer Bewegung rauf fahren)

Ansteuerung des Sauger

- Muss angesteuert werden beim Aufheben
- abgesteuert werden beim Fallen Lassen in die Box

Code Ausschnitte:

Funktion.src

```
EXT SET_PART(point:E6POS);
EXT GET_PART(point:E6POS);
EXT CLAMP();
EXT UNCLAMP();
EXT REFILL();

DEFFCT BOOL SET_PART()
    SPTP Xreturn;
    SLIN XreturnDown;
    ;absaugen
    SLIN Xreturn;
    SPTP XHOME;
    RETURN(TRUE);
ENDFCT

DEFFCT BOOL GET_PART(point:E6POS, apoint:E6POS)
    E6POS point;
    E6POS apoint;
    SPTP apoint;
    SLIN point;
    ;ansaugen
    SLIN apoint;
    SPTP XHOME;
    RETURN TRUE;
```

```

ENDFCT
DEFFCT BOOL CLAMP()
    IF $IN[30] and not $IN[31]
        $OUT[38] = TRUE;
        $OUT[39] = TRUE;
        IF ((NOT $IN[41]) AND ($IN[40])) OR (NOT $IN[39] and $IN[38]) THEN
            return(FALSE);
        ENDIF
    ELSE
        return FALSE
    ENDIF
    return(TRUE);
ENDFCT
DEFFCT BOOL UNCLAMP()
    IF $IN[30] and not $IN[31]
        $OUT[38] = FALSE;
        $OUT[39] = FALSE;
        IF (( $IN[41]) AND (NOT $IN[40])) OR ( $IN[39] and NOT $IN[38]) THEN
            return(FALSE);
        ENDIF
    ELSE
        return FALSE
    ENDIF
    return(TRUE);
ENDFCT
DEFFCT BOOL REFILL()
    if UNCLAMP() THEN
        $OUT[43] = FALSE;
        $OUT[42] = TRUE;
        WAIT SEC 2;
        $OUT[42] = FALSE;
        WAIT SEC 5;
        $OUT[43] = TRUE;
        $OUT[42] = TRUE;
        WAIT SEC 2;
        $OUT[42] = FALSE;
        IF not CLAMP()
            return FALSE
        ENDIF
        return(true);
    ENDIF
    return false;
ENDFCT

```

Main4c.scr

```

&ACCESS
&REL2
&PARAM=*
&PARAM=:
&PARAM=: 4
DECL B00L
DECL B00L
DECL B00L

```

```
DECL BOOL
```

```
EXTSET_PART(:E6POS);
```

```
EXTGET_PART(:E6POS);
```

```
EXTCLAMP();
```

```
EXTUNCLAMP();
```

```
EXTREFILL();
```

```
DEFMain4c()
```

```
;FOLD INI;%{PE}
```

```
;FOLD BASISTECH INI
```

```
GLOBALINTERRUPTDECL 3 ==TRUEIR_STOPM()
```

```
INTERRUPTON3
```

```
BAS(,0)
```

```
;ENDFOLD (BASISTECH INI)
```

```
;FOLD USER INI
```

```
;Make your modifications here
```

```
;ENDFOLD (USER INI)
```

```
;ENDFOLD (INI)
```

```
;FOLD SPTP HOME Vel=100 % DEFAULT ;%{PE}
```

```
;FOLD Parameters ;%{h}
```

```
;Params IlfProvider=kukaroboter_basistech_inlineforms_movement_spline;
```

```
Kuka.IsGlobalPoint=False; Kuka.PointName=HOME; Kuka.BlendingEnabled=False;
```

```
Kuka.MoveDataPtpName=DEFAULT; Kuka.VelocityPtp=100;
```

```
Kuka.VelocityFieldEnabled=True; Kuka.CurrentCDSetIndex=0;
```

```
Kuka.MovementParameterFieldEnabled=True; IlfCommand=SPTP
```

```
;ENDFOLD
```

```
SPTP[1]=SVEL_JOINT(100.0),=STOOL2(),=SBASE(.),=SIPO_MODE(.),=SLOAD(.),
```

```
[1]=SACC_JOINT(),=SAPO_PTP(),[1]=SGEAR_JERK(),[1]=USE_CM_PRO_VALUES(0)
```

```
;ENDFOLD
```

```
=TRUE;
```

```
WHILE
```

```
WHILE
```

```
IFIN[32]THEN
```

```
=FALSE;
```

```
=FALSE;
```

```
ENDIF
```

```
IFNOTIN[33]THEN
```

```
=FALSE;
```

```
ENDIF
```

```
ENDWHILE
```

```
IFTHEN
```

```
IFIN[34]THEN
```

```
IFTHEN
```

```
REFILL();
```

```
=FALSE;
```

```
ENDIF
```

```
CLAMP();
```

```
GET_PART();
```

```
SET_PART();
```

```

GET_PART();
SET_PART();
=TRUE;
ELSEIF IN[35]
IF THEN
;
=FALSE;
ENDIF
CLAMP();
GET_PART();
SET_PART();
GET_PART();
SET_PART();
GET_PART();
SET_PART();
=TRUE;
ENDIF
ENDIF
=TRUE;
ENDWHILE

;FOLD SPTP HOME Vel=100 % DEFAULT ;%{PE}
;FOLD Parameters ;%{h}
;Params IlfProvider=kukaroboter_basistech_inlineforms_movement_spline;
Kuka.IsGlobalPoint=False; Kuka.PointName=HOME; Kuka.BlendingEnabled=False;
Kuka.MoveDataPtpName=DEFAULT; Kuka.VelocityPtp=100;
Kuka.VelocityFieldEnabled=True; Kuka.CurrentCDSetIndex=0;
Kuka.MovementParameterFieldEnabled=True; IlfCommand=SPTP
;ENDFOLD
SPTP[1]=SVEL_JOINT(100.0),=STOOL2(),=SBASE(_),=SIPO_MODE(_),=SLOAD(_),
[1]=SACC_JOINT(),=SAPO_PTP(),[1]=SGEAR_JERK(),[1]=USE_CM_PRO_VALUES(0)
;ENDFOLD

END

```

"Funktion.src:" sind die Hilfsfunktionen die wir verwenden

SET_PART: Geht von Home zur Box

GET_PART Nimmt die Teile von der Pallet geht zu home

CLAMP: Fixiert die Pallette

UNCLAMP: Lässt die Pallette aus

REFILL Fährt hin und zurück

"Main4c.src": führt das eigentliche Programm aus

Der Code spricht für sich selbst

Erneute Definierung der Punkte, durch Verschiebung von Sauger unten

- Punkte Oben und Unten
- Uploaden des Codes von Saugen und Entsaugen

Sources

<https://elearning.tgm.ac.at/mod/assign/view.php?id=97192>

<https://app.diagrams.net/>