

Luis Winkler



Ergänzungen zur Projektdokumentation



ROBODRINKS

Schule: BBS1-Mainz
Schulform: Fachschule
Klasse: FSAPZ18
Modul: 15
Zeitraum: 06.2021-07.02.2022
Betreuer: Herr Markus Musielack / Herr Guido Müller
Abgabetermin: 17.02.2022

Inhaltsverzeichnis

1	Gläserweiche	2
2	Roboterprogrammierung	3
3	Magazin SPS-Programmierung	4
3.1	Erklärung Aufbau und Funktion	4
3.1.1	Einfacher Pfad	5
3.1.2	Verzweigung eins bis vier	5
3.1.3	Verzweigung fünf	6
3.1.4	Verzweigung sechs und sieben	6
3.1.5	Verzweigung acht	6
4	Fazit	6
5	Anhang	7
6	Eidesstattliche Erklärung	7

1 Gläserweiche

Damit die Gläser am Ende des Rückgabebandes erkannt werden, wurden zwei kapazitive Sensoren an einem Bosch-Profil befestigt. Das Profil wurde mittels eines Winkels auf der Zentriervorrichtung festgeschraubt.

Wenn ein Glas zurückgegeben wird, kommt es vor, dass dieses an die Sensoren bzw. deren Halterung stößt und ein unschönes Geräusch von sich gibt oder im schlimmsten Fall zerbricht.

Natürlich könnte man die Sensoren auch so ausrichten, dass die Gläser diese nie berühren, jedoch würde dies Probleme mit der Positionierung des Roboters ergeben, da er an dieser Stelle an seine Achsgrenzen kommen würde. Man müsste ihn an der Greifposition so ausrichten, dass er die Gläser greift und entnimmt ohne die Sensoren zu berühren. Zudem müssen die Sensoren einen sehr geringen Abstand zu den Gläser haben damit diese erkannt werden.

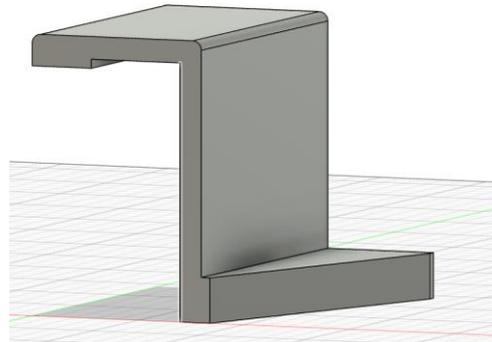


Abb. 1: Gläserweiche

Damit die Gläser an den Sensoren vorbeigeführt werden und auch der Roboter die bestmögliche Position anfahren kann, wurde eine Weiche designt und mittels 3D-Druckers aus schwarzen PLA hergestellt. (siehe Abb. 1)

Dieses Bauteil wurde ca. 30 cm vor der Gläsererkennung an der Plattformkante der Anlage befestigt. (siehe Abb. 2)

Nun wird jedes Glas (Bier, Cocktail & Kaffee) von der Weiche auf dem Förderband zentriert, so dass es an den Sensoren vorbeifährt und sich optimal zur Erkennung am Ende des Förderbandes positioniert.



Abb. 2: Befestigung Gläserweiche

2 Roboterprogrammierung

Nach der Dokumentation die am 10.01.2022 abgegeben wurde, hat sich noch einiges an der Anlage getan. Gerade der Roboter war hier oft im Mittelpunkt.

Als das Magazin fertiggestellt wurde, mussten die Teachpunkte für den Roboter noch angelernt werden. (siehe Abb. 3)

Darüber hinaus kamen weitere Hilfspunkte hinzu, um Kollisionen zu verhindern.

Am Roboterprogramm wurde ebenso die eine oder andere Zeile hinzugefügt bzw.

verbessert. Z.B. der Befehl

OVRD, welcher dazu dient, die Geschwindigkeit des Roboters zu definieren, da man mehrere Geschwindigkeiten benötigt je nachdem was für Positionen man anfährt. Um nicht im ganzen Programm alle Befehle anzupassen wenn es zu einer anderen Wahl der Geschwindigkeit kommt, wurden im Hauptprogramm drei Variablen vom Typ Integer definiert (G1, G2, G3).

In jeder Variable wird eine andere Geschwindigkeit gespeichert, nun müssen diese Variablen an den Befehl OVRD angehängt werden (z.B. OVRD G1). Der Wert der Variablen wird nur einmal im Hauptprogramm angepasst. (siehe Abb. 4)

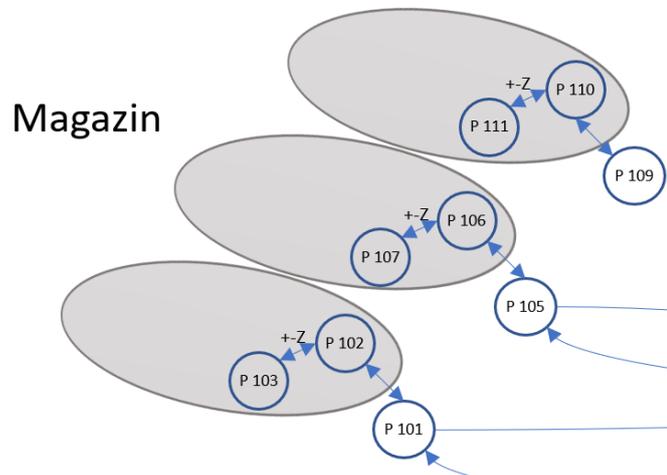


Abb. 3: Magazin Teachpunkte

```

C:\PROGRAM FILES (X86)\COSIROP\PROJECTS\ROBODRINKS\1.MB4
1 DEF INTE G1 '30% der Max. Geschwindigkeit
3 DEF INTE G2 '60% der Max. Geschwindigkeit
6 DEF INTE G3 '90% der Max. Geschwindigkeit
9 G1 = 30
12 G2 = 60
15 G3 = 90
40 ACCEL 100,100
43 OVRD G3
45 HCLOSE 1
50 MOV P350
53 M_OUT(10)=0 'Roboter nicht mehr Busy
60 IF (M_IN(7)=1)THEN
70 GOSUB *KAGET
80 ENDIF
90 IF (M_IN(9)=1) THEN
94 GOSUB *BIER
97 ENDIF
100 IF (M_IN(8)=1) THEN
110 GOSUB *COCKDO
130 ENDIF
140 IF (M_IN(6)=1) THEN
150 GOSUB *KADO
170 ENDIF
180 IF (M_IN(13)=1)THEN
190 GOSUB *SPUEGG
210 ENDIF
220 IF (M_IN(14)=1) THEN
230 GOSUB *SPUEKG
240 ENDIF
250 GOTO 1

```

Abb. 4: Roboter Hauptprogramm

Dies ergab eine sehr große Zeiterspannis bei der Inbetriebnahme der Anlage, da man sehr schnell den Roboter auf andere Geschwindigkeiten fahren lassen konnte.

3 Magazin SPS-Programmierung

3.1 Erklärung Aufbau und Funktion

Die Hauptaufgabe des Teams war der Bau eines intelligenten Lagersystems. Das SPS Programm musste also auch dementsprechend angepasst bzw. erweitert werden.

Es wurde eine neue POE (Programmorganisationseinheit) erstellt, mit dem Namen Magazin (siehe Abb. 5). Man entschied sich bei der Wahl der Programmiersprachen für die AS (Ablaufsprache) und ST (Strukturierter Text).

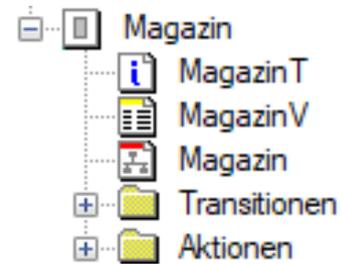


Abb. 5: POE

Die Ablaufsprache erwies sich als hilfreich für die Hauptstruktur (siehe Abb. 6), da man schnell erkennen kann, wie der Verlauf der Funktion ist. Man hat sich auch

die Besonderheit von der Software PC-WORX zu Nutze gemacht, jede Transition und Aktion in einer Lupe ausführen zu können. Diese Lupen wurden alle mit dem strukturierten Text programmiert. Der Vorteil davon ist, dass sehr komplexe mathematische Funktionen mit wenig Text erstellbar sind.

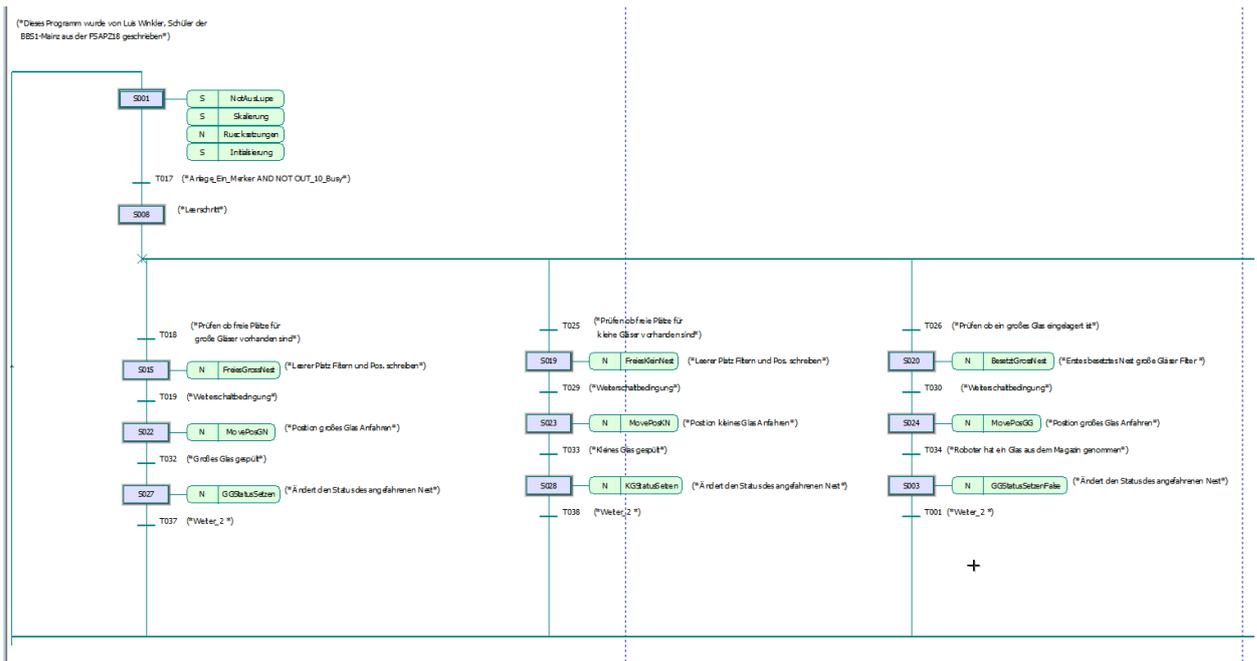


Abb. 6: Programmstruktur

Das Programm Magazin besteht aus einem einfachen Pfad gefolgt von acht alternativ Zweigen, in denen unterschiedliche Funktionen abgearbeitet werden. Hier eine Auflistung mit der entsprechenden Erklärung:

3.1.1 Einfacher Pfad

Der einfache Pfad besteht aus zwei Schritten und einer Transition (siehe Abb. 7), wobei der zweite Schritt ein Leerschritt ist.

(*Dieses Programm wurde von Luis Winkler, Schüler der BBS1-Mainz aus der FSAPZ18 geschrieben*)

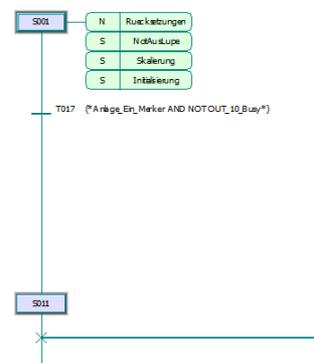


Abb. 7: Einfacher Pfad

Im ersten Schritt sind vier Lupen angehängt:

- **Lupe Rücksetzungen:** Ist nur aktiv, wenn erster Schritt aktiv ist. Hier werden Variablen gesetzt oder rückgesetzt, je nachdem wie es die Funktion verlangt.
- **Lupe Not-Aus:** Immer aktiv. Hier wird das Magazin bei Betätigung des Not-Haltkreises gestoppt, und bei Auswahl im Bedienfeld („Anlage Aus“) kann auch die Schrittkette des Magazins zurückgesetzt werden.
- **Skalierung:** Immer aktiv. Hier wird die Skalierung der Positionen der Nester vorgenommen. Aus den Werten „MagGeberWert“ und „MagGeberUmdrehungen“ werden eigene Positionen erstellt, damit die Nester angefahren werden können.
- **Initialisierung:** Immer aktiv. Hier werden mehrere Werte initialisiert, wie z.B. Teachpunkte des Magazins als auch die Erkennung der besetzten Nester wie auch freie Nester.

3.1.2 Verzweigung eins bis vier

Die ersten vier Verzweigungen haben denselben Aufbau, nur die Funktion ist leicht unterschiedlich. Bei der ersten wird ein leeres Bierglasnest gefiltert, dessen Position angefahren und den Status neu geschrieben. Bei der zweiten Verzweigung geschieht das gleiche, aber für die Cocktail bzw. Kaffeegläser. Bei der dritten und vierten Verzweigung passiert das Gegenteil wie bei der ersten und zweiten. Es werden die besetzten Nester gefiltert, angefahren und der Status verändert.

Um das Ganze mit möglichst wenig Programmieraufwand zu realisieren, wurde beim Filtern der Nester eine FOR-Schleife (siehe Abb. 8) verwendet.

```
(*Auswertung was für ein Nest der großen Gläser frei ist, und
Schreiben der Teachposition in die Zielpositions Variable*)

MagazinBereit:=TRUE;

FOR i := 31 TO 40 BY 1
  DC
  IF NOT MagStatus[i] THEN
    MagazinZielPos := MagPosition[i];
    Weiter_1:=TRUE;
  END_IF;
END_FOR;
```

Abb. 8: FOR-Schleife

3.1.3 Verzweigung fünf

Diese Verzweigung ist für das automatische Zählen des Gläserbestand im Magazin. Als erstes wird die Grundstellung angefahren, danach erneut im Uhrzeigersinn bis zur rechten Grenze, dabei erkennen Kapazitive Sensoren auf den drei Etagen ob die jeweiligen Nester belegt oder frei sind. Der Status wird gespeichert und im HMI visualisiert. (näheres zur automatischen Gläserzählung im Dokumentationsteil von Marko Parlov)

3.1.4 Verzweigung sechs und sieben

In diesen zwei Verzweigungen ist es möglich das Magazin händisch im Rechtslauf als auch im Linkslauf drehen zu lassen. Dafür im HMI auf das entsprechende Symbol klicken. (siehe Abb. 9)



Abb. 9: Rechts-Linkslauf Schaltfläche

3.1.5 Verzweigung acht

Hiermit ist es den Betreuer der Anlage möglich eine beliebige Position des Magazins anzufahren. Für

Inbetriebnahme Zwecke bietet sich diese Funktion hervorragend an, leider ist dies nur mit der Debugg Funktion der Software PC-WORX möglich. Integriert man die benötigten Variable in das HMI, ist eine Bedienung auch von hier aus möglich.

Um die gewünschte Position anzufahren, muss lediglich die Variable „HMINestPos“ vom Typ INT auf einen Wert von 1 bis 40 gesetzt werden. Dann die Variable „HMIGoPos“ vom Typ BOOL auf TRUE, das Magazin fährt an die gewünschte Position. (siehe Abb. 10)

HMIGoPos	BOOL	VAR_EXTERNAL	Fahre das gewählte Nest an
HMINestPos	INT	VAR_EXTERNAL	Gibt man eine Zahl von 1..40 an, kann man das dazugehörige Nest anfahren

Abb. 10: Variablen

4 Fazit

Nach der Abgabe der Hauptdokumentation am 10.01.2022, ist an der Anlage vieles passiert. So wurde das Magazin, das dazugehörige SPS-Programm, das Roboterprogramm und viele kleinere Baustellen fertiggestellt.

Da nicht mehr viel Zeit bis zur Projektabgabe zu Verfügung stand, konnte man etliche Probleme, gerade in der Programmierung, nicht perfekt lösen, da meist viel Zeit benötigt wird, um geschriebene Codes zu testen. Trotzdem sind die programmierten Funktionen sehr gute Lösungen geworden.

Umso größer eine Anlage ist, gerade im Falle des Getränkeautomaten, der aus vielen Komponenten besteht, je höher der Grad an Instandsetzung. Benötigt ein Team viel Zeit und Ressourcen, um den Ist-Zustand funktionsfähig zu machen, so bleibt oftmals wenig Zeit, um die Aufgaben zu realisieren, die neu assigniert wurden.

Als Schlusswort gilt zu sagen, dass das Projekt genau die Fähigkeiten abrufte, die ein Prozessautomatisierungstechniker beherrschen sollte. Oftmals stößt man an Technischen Grenzen, die aber im Team in der Regel schnell gelöst werden können.

Auf die nachfolgenden Teams wartet ein anspruchsvolles Projekt, mit vielen Herausforderungen und Möglichkeiten all das Wissen aus der Technikerschule als auch aus dem privaten Umfeld anzuwenden.

5 Anhang

- Roboterprogramm RoboDrinks inkl. Programmabläufe
- CAD-Zeichnung

Anhänge in digitaler Form auf der beigefügten CD

- \SPS-Programm\RoboDrinks SPS-Programm.zip
- \Roboterprogramm RoboDrinks\Roboterprogramm RoboDrinks.zip
- \Sonstiges\Gläserweiche.pdf
- \CAD-Dateien\Gläserweiche.f3d

6 Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Hausarbeit selbständig verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe. Die Stellen der Hausarbeit, die anderen Quellen im Wortlaut oder dem Sinn nach entnommen wurden, sind durch Angaben der Herkunft kenntlich gemacht. Dies gilt auch für Zeichnungen, Skizzen, bildliche Darstellungen sowie für Quellen aus dem Internet.

Mainz, den 17.02.2022

Nachname: _____

Vorname: : _____

Unterschrift: : _____