



RT
Bericht

2021

Regelungstechnik

30. November 2021

Florian Tietjen 2519584

Emily Antosch 2519935

INHALTSVERZEICHNIS

Abbildungsverzeichnis	3
1 SIPN mit ST	4
1.1 Einführung	4
1.2 Vorbereitung	4
1.3 Programmierung der Anlage in ST durch SIPN	7

ABBILDUNGSVERZEICHNIS

1.1 Petrinetz für den Automatikmodus der Anlage 5

1.1 Einführung

Im nächsten Teil des Labors wollen wir uns mit dem Lager und dem Transportarm beschäftigen. Dieses Mal verwenden wir dafür allerdings die Programmiersprache ST und das Prinzip von Steuerungstechnisch interpretierten Petrinetzen (SIPN). Wir wollen einen steuerungstechnischen Ablauf erstellen, bei dem in einer dauerhaften Schleife, die Werkstücke vom Lager zum Band transportiert werden, um dort weiter verarbeitet zu werden.

1.2 Vorbereitung

Wir wollen uns zunächst mit dem Petrinetz befassen, welches wir dann dazu verwenden wollen, einen Teil der Steuerungsstrecke in ST zu programmieren. Das Petrinetz enthält, wie in den Anforderungen genannt, nur den Automatikmodus der Anlage. Der Betriebskopf zum Umschalten wird erst im folgenden Praktikum behandelt und ist daher nur schematisch durch Schreiben der Variablen beim Debuggen der Anlage in CodeSYS umgesetzt.

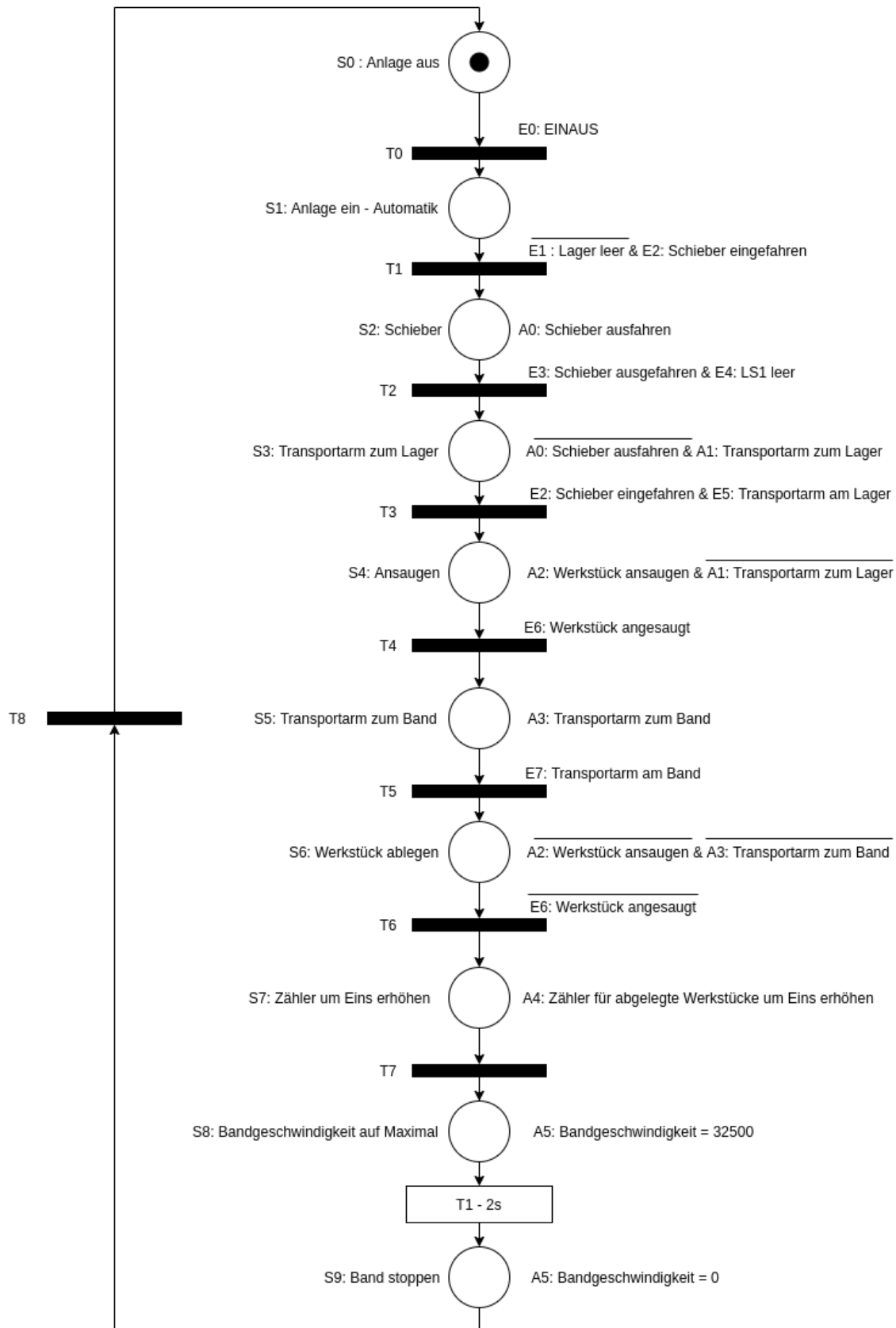


Abbildung 1.1: Petrinetz für den Automatikmodus der Anlage

Wie oben zu sehen ist, haben wir den Ablauf der technischen Anforderungen umgesetzt. In einer laufenden Schleife, wird zunächst der Schieber mit einem Werkstück ausgefahren und dann von dem Transportarm abgeholt, solange in der Pufferstrecke Platz ist. Im Anschluss saugt der Arm das Werkstück an und transportiert dieses zum Band, um es dort wieder loszulassen. Der Zähler für die abgelegt, das Band wird für 2 Sekunden auf der maximalen Geschwindigkeit bewegt und kommt dann zum Stehen, woraufhin der Ablauf von vorne anfängt. Wurde an irgendeinem Punkt die EINAUS-Funktion auf AUS gestellt, wird Sie genau an diesem Punkt umgesetzt und die Anlage ist außer Betrieb.

1.3 Programmierung der Anlage in ST durch SIPN

Wir haben den folgenden Code in ST für die Steuerung der Anlage verwendet

```

1  VAR_INPUT
2      ls1: BOOL; (* Lichtschranke 1 leer *)
3      piece_sucked: BOOL; (* Werkstueck angesaugt *)
4      storage_empty: BOOL; (* Lagerplatz leer *)
5      piston_extended: BOOL; (* Piston ausgeklappt *)
6      piston_reduced: BOOL; (* Piston eingeklappt *)
7      arm_at_storage: BOOL; (* Arm am Lagerplatz *)
8      arm_at_band: BOOL; (* Arm am Band *)
9  END_VAR
10 VAR_OUTPUT
11      band_speed: WORD; (* Bandgeschwindigkeit auf Hardware *)
12      extend_piston : BOOL (* Ausschieber auf Hardware ausfahren *)
13      arm_to_lager : BOOL (* Arm zum Lager auf Hardware fahren *)
14      arm_to_band : BOOL (* Arm zum Band auf Hardware fahren *)
15      suck_piece : BOOL (* Werkstueck angesaugt *)
16      release_piece : BOOL (* Werkstueck loslassen *)
17 END_VAR
18 VAR
19     S0 : BOOL := TRUE (* Initialisierung des Startzustands *)
20     S1, S2, S3, S4, S5, S6, S7, S8, S9 : BOOL; (* Zustaende *)
21     counter : WORD; (* Zaehler fuer abgelegte Werkstuecke *)
22     counter_FP : BOOL; (* Flankendetektion fuer Zaehler *)
23     counter_FHM : BOOL; (* Flankendetektion fuer Zaehler *)
24     EINAUS: BOOL; (* Ein/Aus *)
25     AUTOMAN: BOOL := 1; (* Automatisch/Manuell *)
26     HAND_BAND_EIN: BOOL; (* Transportband bewegen/stoppen *)
27     HAND_BANDV: WORD; (* Transportband Geschwindigkeit *)
28     HAND_AUSSCHIEBER: BOOL; (* Ausschieber aktiv *)
29     HAND_ARM_ZUM_LAGER: BOOL; (* Arm zum Lager *)
30     HAND_ARM_ZUM_BAND: BOOL; (* Arm zum Band *)
31     HAND_ANSAUGEN: BOOL; (* Werkstueck ansaugen *)
32     HAND_LOSLASSEN: BOOL; (* Werkstueck loslassen *)
33     T1 : TON; (* Timer fuer Band *)
34 END_VAR
35 IF AUTOMAN THEN
36     WHILE EINAUS DO
37         IF (S0 AND NOT S1 AND EINAUS) THEN
38             S1 := TRUE;
39             S0 := FALSE;
40         END_IF;
41         IF (S1 AND NOT S2 AND NOT storage_empty AND piston_reduced) THEN
42             S2 := TRUE;
43             S1 := FALSE;
44         END_IF;
45         IF (S2 AND NOT S3 AND piston_extended AND ls1) THEN
46             S3 := TRUE;
47             S2 := FALSE;
48         END_IF;
49         IF (S3 AND NOT S4 AND arm_at_storage AND piston_reduced) THEN
50             S4 := TRUE;
51             S3 := FALSE;
52         END_IF;
53         IF (S4 AND NOT S5 AND piece_sucked) THEN
54             S5 := TRUE;
55             S4 := FALSE;
56         END_IF;
57         IF (S5 AND NOT S6 AND arm_at_band) THEN
58             S6 := TRUE;
59             S5 := FALSE;
60         END_IF;
61         IF (S6 AND NOT S7 AND NOT piece_sucked) THEN
62             S7 := TRUE;

```

```

63     S6 := FALSE;
64 END_IF;
65 IF (S7 AND NOT S8 AND T1.Q) THEN
66     S8 := TRUE;
67     S7 := FALSE;
68     band_speed := 32500;
69 END_IF;
70 IF (S8 AND NOT S9) THEN
71     S9 := TRUE;
72     S8 := FALSE;
73     band_speed := 0;
74 END_IF;
75 IF (S9 AND NOT S0) THEN
76     S0 := TRUE;
77     S9 := FALSE;
78 END_IF;
79 extend_piston := S2;
80 arm_to_lager := S3;
81 suck_piece := S4 OR S5;
82 arm_to_band := S5;
83 release_piece := S6;
84 counter_FP := S7 AND NOT counter_FHM;
85 counter_FHM := S7;
86 IF counter_FP THEN
87     counter := counter + 1;
88 END_IF;
89 IF S8 THEN
90     band_speed := 32500;
91 ELSE
92     band_speed := 0;
93 END_IF;
94 T1(IN:=S7 AND NOT S8, PT:=T#2s);
95 END_WHILE;
96 ELSE
97     IF EINAUS THEN
98         IF HAND_BAND_EIN THEN
99             band_speed := HAND_BANDV;
100         ELSE
101             band_speed := 0;
102         END_IF;
103
104         IF HAND_AUSSCHIEBER THEN
105             extend_piston := TRUE;
106         ELSE
107             extend_piston := FALSE;
108         END_IF;
109
110         IF HAND_ARM_ZUM_LAGER AND NOT HAND_ARM_ZUM_BAND THEN
111             arm_to_lager := TRUE;
112         ELSE
113             arm_to_lager := FALSE;
114         END_IF;
115
116         IF HAND_ARM_ZUM_BAND AND NOT HAND_ARM_ZUM_LAGER THEN
117             arm_to_band := TRUE;
118         ELSE
119             arm_to_band := FALSE;
120         END_IF;
121
122         IF HAND_ANSAUGEN AND NOT HAND_LOSLASSEN THEN
123             suck_piece := TRUE;
124         ELSE
125             suck_piece := FALSE;
126         END_IF;
127
128         IF HAND_LOSLASSEN AND NOT HAND_ANSAUGEN THEN
129             release_piece := TRUE;
130         ELSE
131             release_piece := FALSE;
132         END_IF;
133     END_IF;
134 END_IF;

```