

Praktikum Ereignisdiskrete Systeme  
S1 - Türsteuerung (Handbetrieb)

Gruppe 13 (Sven Richter, Mathias Thom)

12.05.2004

# 1 Signalbeschreibung

---

Sensoren:	(High aktiv)
X1	Endlageschalter → offen
XZ	Endlageschalter → geschlossen
EO (E1 ∧ E3)	Schalter öffnen
ES (E2 ∧ E4)	Schalter schließen

---

Aktoren:	(low aktiv)
Y1	Antrieb "Tür öffnen"
Y2	Antrieb "Tür schließen"
Y3	Warnlampe

---

- Aktivierung durch kurzes betätigen der Schalter
- Öffner dominieren über Schließer
- während des Schließens erneutes Öffnen möglich
- erst nach vollständigem Öffnen ist erneutes Schließen möglich

## 2 Schaltbelegungstabelle

EO	ES	X1	XZ	Y3	Y2	Y1	Zustand	Erläuterung
0	0	0	0	0	0	0	F	beide Endlage aktiv
0	0	0	1	0	0	0	1	beide Schalter, Tür Offen
0	0	1	0	0	0	1	2	beide Schalter, Tür Zu
0	0	1	1	0	0	1	3	beide Schalter, Tür öffnet
0	1	0	0	0	0	0	F	beide Endlage aktiv
0	1	0	1	0	0	0	4	öffnen Schalter, Tür Offen
0	1	1	0	0	0	1	5	öffnen Schalter, Tür Zu
0	1	1	1	0	0	1	6	öffnen Schalter, Tür öffnet
1	0	0	0	0	0	0	F	beide Endlage aktiv
1	0	0	1	1	1	0	7	schließen Schalter, Tür Offen
1	0	1	0	0	0	0	8	schließen Schalter, Tür Zu
1	0	1	1	0	0	1	9	schließen Schalter, Tür öffnet
1	0	1	1	1	1	0	10	schließen Schalter, Tür schließt
1	1	0	0	0	0	0	F	beide Endlage aktiv
1	1	0	1	0	0	0	11	kein Schalter, Tür Offen
1	1	1	0	0	0	0	12	kein Schalter, Tür Zu
1	1	1	1	0	0	1	13	kein Schalter, Tür öffnet
1	1	1	1	1	1	0	14	kein Schalter, Tür schließt

### 3 Automatentabelle

Vektor: EO ES X1 XZ / Y3 Y2 Y1

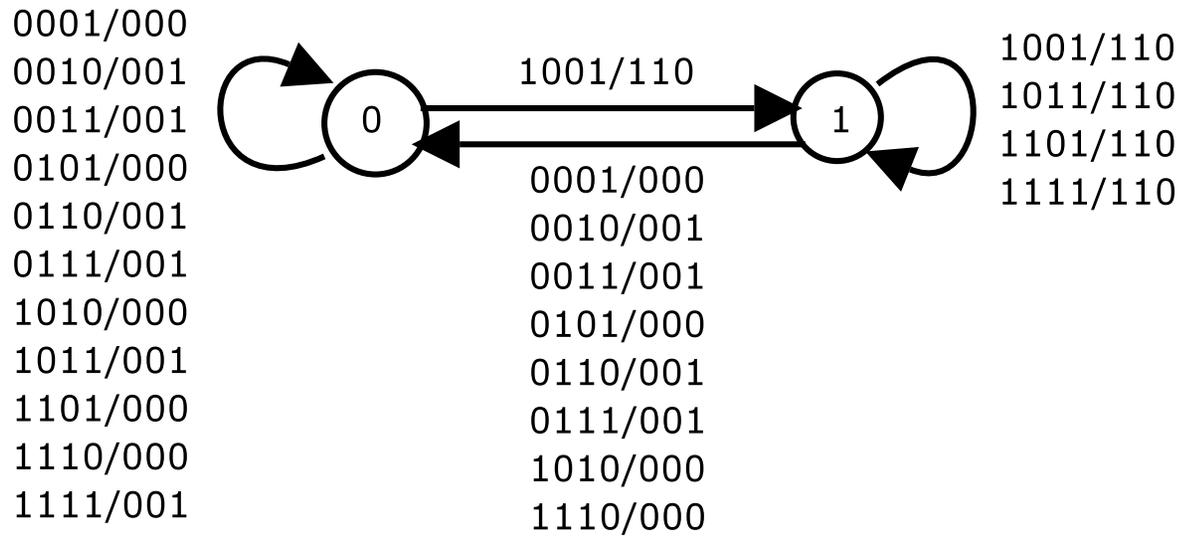
Z	0001	0010	0011	0101	0110	0111	1001	1010	1011	1101	1110	1111
1	1/000			4/000			10/110			11/000		
2		3/001	3/001		6/001	6/001		8/000	9/001		12/000	13/001
3	1/000		3/001	4/000		6/001	10/110		9/001	11/000		13/001
4	1/000			4/000			10/110			11/000		
5		3/001	3/001		6/001	6/001		8/000	9/001		12/000	13/001
6	1/000		3/001	4/000		6/001	10/110		9/001	11/000		13/001
7	1/000		3/001	4/000		6/001	10/110		10/110	14/110		14/110
8		3/001			6/001			8/000			12/000	
9	1/000		3/001	4/000		6/001	10/110		9/001	11/000		13/001
10		3/001	3/001		6/001	6/001		8/000	10/110		12/000	14/110
11	1/000			4/000			10/110			11/000		
12		3/001			6/001			8/000			12/000	
13	1/000		3/001	4/000		6/001	10/110		9/001	11/000		13/001
14		3/001	3/001		6/001	6/001		8/000	10/110		12/000	14/110

### 4 reduzierte Automatentabelle

Vektor: EO ES X1 XZ / Y3 Y2 Y1

Z	0001	0010	0011	0101	0110	0111	1001	1010	1011	1101	1110	1111
A/0	0/000	0/001	0/001	0/000	0/001	0/001	1/110	0/000	0/001	0/000	0/000	0/001
B/1	0/000	0/001	0/001	0/000	0/001	0/001	1/110	0/000	1/110	1/110	0/000	1/110

## 5 reduzierter Automatengraph



## 6 Erweiterte Schaltbelegungstabelle

$Z^{(k)}$	EO	ES	X1	XZ	$Z^{(k+1)}$	s1	r1	Y3	Y2	Y1
0	0	0	0	1	0	0		0	0	0
0	0	0	1	0	0	0		0	0	1
0	0	0	1	1	0	0		0	0	1
0	0	1	0	1	0	0		0	0	0
0	0	1	1	0	0	0		0	0	1
0	0	1	1	1	0	0		0	0	1
0	1	0	0	1	1	1	0	1	1	0
0	1	0	1	0	0	0		0	0	0
0	1	0	1	1	0	0		0	0	1
0	1	1	0	1	0	0		0	0	0
0	1	1	1	0	0	0		0	0	0
0	1	1	1	1	0	0		0	0	1
1	0	0	0	1	0	0	1	0	0	0
1	0	0	1	0	0	0	1	0	0	1
1	0	0	1	1	0	0	1	0	0	1
1	0	1	0	1	0	0	1	0	0	0
1	0	1	1	0	0	0	1	0	0	1
1	0	1	1	1	0	0	1	0	0	1
1	1	0	0	1	1	1	0	1	1	0
1	1	0	1	0	0	0	1	0	0	0
1	1	0	1	1	1	1	0	1	1	0
1	1	1	0	1	1	1	0	1	1	0
1	1	1	1	0	0	0	1	0	0	0
1	1	1	1	1	1	1	0	1	1	0

## 7 Boolesche Gleichungen

$$\begin{aligned}Z^{k+1} &= \bar{Z} EO \bar{ES} \bar{X1} XZ \vee Z EO (\bar{ES}(X1 \vee \bar{X1}) \vee ES(X1 \vee \bar{X1})) XZ \\ &= EO XZ (\bar{Z} \bar{ES} \bar{X1} \vee Z)\end{aligned}$$

$$s1 = \bar{Z} EO \bar{ES} \bar{X1} XZ$$

$$r1 = Z(\bar{XZ} \vee \bar{EO})$$

$$Y3 = Y2 = Z EO XZ \vee EO \bar{ES} \bar{X1} XZ$$

$$Y1 = \bar{EO} X1 \vee \bar{Z} X1 XZ$$

## 8 Programmimplementation

### 8.1 Durchlauf

Das Programm lief ohne Probleme. Korrekturen am Versuchsplatz waren nicht nötig.

### 8.2 Quelltext der Funktion manuell

```
void manuell(void) {

    int xz = kanaele.ein.X0;
    int x1 = kanaele.ein.X1;
    int eo = kanaele.ein.E1 & kanaele.ein.E3;
    int es = kanaele.ein.E2 & kanaele.ein.E4;

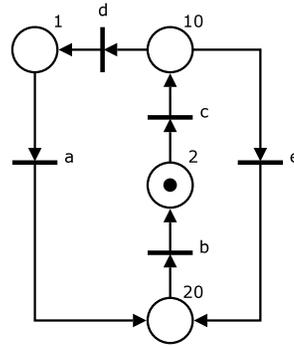
    int yo,ys;

    // wenn open und close gleichzeitig .. schliessen loeschen
    if (!eo & !es) {
        es = 1;
    }

    // wenn beide Entschalter . FEHLER
    if (!xz & !x1) {
        yo = 0;
        ys = 0;
        printf ("FEHLER - beide Entschalter aktiv\n");
    } else {
        zustand = eo & xz & ((!zustand & !es & !x1) | zustand);
        ys = (zustand & eo & xz )|(eo & !es & !x1 & xz);
        yo = (!eo & x1) | (!zustand & x1 & xz);
    }

    ausgang = yo + (ys << 2) + (ys << 1);
    DIO_Write (ausgang);
}}
```

## 9 alternative Lösung durch Petri-Netz



Plätze:

- 1 Tür ist geschlossen (Ausgangsbelegung:  $Y1=Y2=Y3=0$ )
- 2 Tür ist offen (Ausgangsbelegung:  $Y1=Y2=Y3=0$ )
- 10 Tür ist in Bewegung und schließt (Ausgangsbelegung:  $Y1=0, Y2=Y3=1$ )
- 20 Tür ist in Bewegung und öffnet (Ausgangsbelegung:  $Y1=1, Y2=Y3=0$ )

Transitionen:

- a Öffnen Schalter wir betätigt ( $EO=0$ )
- b Türementschalter "offen" aktiv ( $X1=0$ )
- c Schliesen Schalter wird betätigt ( $ES=0$ )
- d Türementschalter "geschlossen" ( $XZ=0$ )
- e Öffnen Schalter wir betätigt ( $EO=0$ )

Der Fehlerzustand (9) wurde im Petri-Netz der Übersicht wegen nicht dargestellt, da das Erreichen des selbigen jederzeit möglich ist, wenn beide Türementschalter gleichzeitig auslösen ( $X1=XZ=0$ ). In diesem Zustand wird die Tür gestoppt (Ausgangsbelegung:  $Y1=1, Y2=Y3=0$ ).

## 9.1 Durchlauf

Bei dieser Lösung ist der Quelltext im Vergleich zu den Booleschen Gleichungen bedeutend länger und benötigt natürlich mehr Speicher allerdings wird er auch sehr viel Übersichtlicher und Korrekturen sind ohne größere Probleme möglich.

Auch diese Funktion lief sofort ohne weitere Probleme.

## 9.2 Quelltext der Funktion manuell

```
void manuell(void) {

    int xz = kanaele.ein.X0;
    int x1 = kanaele.ein.X1;
    int eo = kanaele.ein.E1 & kanaele.ein.E3;
    int es = kanaele.ein.E2 & kanaele.ein.E4;

    int yo,ys;

    // wenn open und close gleichzeitig .. schliessen loeschen
    if (!eo & !es) {
        es = 1;
    }

    // wenn beide Entschalter . FEHLER
    if (!xz & !x1) {
        zustand =9;
    }

    switch (zustand) {
        case 1: // geschlossen
            yo=0;ys=0; //init
            if (!eo) zustand = 20;
            break;
        case 2: // offen
            yo=0;ys=0; //init
            if (!es) zustand = 10;
    }
}
```

```
        break;
    case 10: //schliesen
        yo=0;ys=1; // init
        if (!xz) zustand = 1;
        if (!eo) zustand = 20;
        break;
    case 20: //oeffnen
        yo=1;ys=0; //init
        if (!x1) zustand = 2;
        break;

    case 9: // fehler
        yo=0;ys=0;
        break;

    default: // zustand rausfinden
        zustand = 20;
        if (ys) zustand = 10;
        if (!xz) zustand = 1;
        if (!x1) zustand = 2;
        break;
}

ausgang = yo + (ys << 2) + (ys << 1);
DIO_Write (ausgang);
}}
```