

EIB System

Funktionsmodul und FMTool

# **FMTool und Funktionsmodul**

## **Technische Dokumentation Beispielkatalog für die Programmierung des Funktionsmoduls**

**Version 1.80**



## **EIB System**

### **Funktionsmodul und FMTool**

#### Hinweise

Die in diesen Unterlagen enthaltenen Angaben, Daten, Werte usw. können ohne vorherige Ankündigung geändert werden. Ebenso sind die Abbildungen unverbindlich.

Ohne ausdrückliche schriftliche Erlaubnis von Gira, Giersiepen GmbH & Co. KG darf kein Teil dieser Unterlagen für irgendwelche Zwecke vervielfältigt oder übertragen werden, unabhängig davon, auf welche Art und Weise und mit welchen Mitteln, elektronisch oder mechanisch, dies geschieht.

©2000 Gira, Giersiepen GmbH & Co. KG  
Alle Rechte vorbehalten.

Alle im Handbuch verwendeten Produktbezeichnungen sind eingetragene Warenzeichen der jeweiligen Firmen.

Gira  
Giersiepen GmbH & Co. KG  
Dahlienstrasse  
D-42477 Radevormwald

Telefon      02195 / 602-0  
Telefax      02195 / 602-118  
Internet      <http://www.gira.de>



## Inhaltsverzeichnis

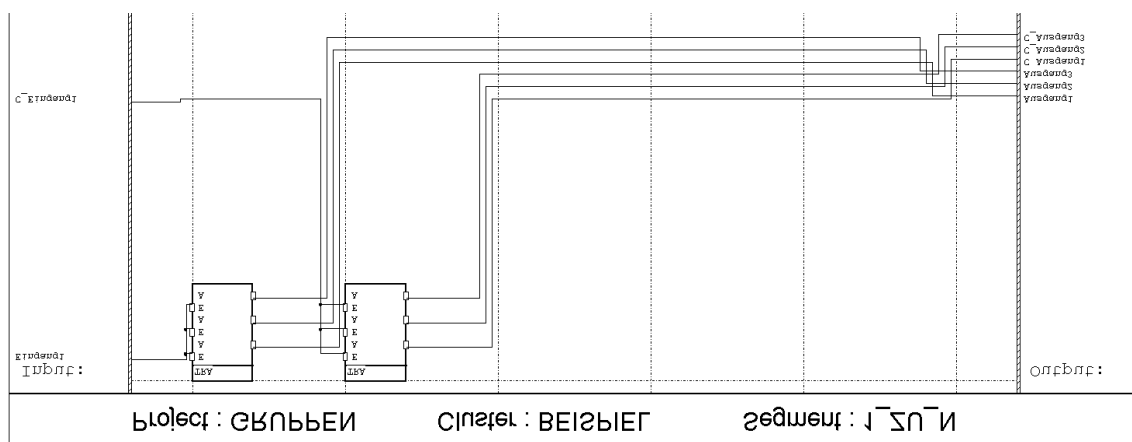
1 SENDEN EINES SIGNALES AUF MEHRERE GRUPPENADRESSEN .....	7
2 STEuern EINES AKTORS MIT MEHREREN SENSOREN.....	8
3 VERGLEICHEN VON WERTEN.....	12
4 DATENTYP-UMWANDLUNG VON BYTE IN VALUE.....	14
5 DATENTYP-UMWANDLUNG VON VALUE IN BYTE.....	15
6 POSITIVE FLANKEN ERKENNEN.....	17
7 NEGATIVE FLANKEN ERKENNEN .....	18
8 ERKENNEN VON LANGEN UND KURZEN TASTENDRÜCKEN .....	19
9 TREPPENHAUSLICHT, EINFACH.....	22
10 TREPPENHAUSLICHT, MIT ÖKOSCHALTUNG.....	24
11 TREPPENHAUSLICHT, MIT DAUERLICHTSCHALTUNG .....	25
12 AUSGABE VON TELEGRAMMEN VERHINDERN.....	30
13 REDUNDANZ MIT ZWEI FUNKTIONSMODULEN.....	31
14 ZEIT SETZEN .....	43
15 DATUM SETZEN.....	46
16 ZYKLISCHE AUSGABE VON WERTEN.....	49
17 ZÄHLER FÜR DATENTYP WORD .....	51
18 ERKENNEN VON DOPPELKLICK.....	53
19 ÄNDERN VON EINSTELLWERTEN ZUR LAUFZEIT .....	56
20 DEBUGGEN.....	59
21 ZEITABHÄNGIGE STEUERUNGEN (SUHR-BLOCK).....	62
22 DATUMSABHÄNGIGE STEUERUNGEN .....	66
23 VERGLEICHEN VON WERTEN MIT HYSTERESE.....	69
24 AUTOMATISCHES BERECHNEN VON ZEITEN BEI FUNKTIONSBLOCKEN.....	71
25 TELEGRAMMFLUSS EIN- UND AUSSCHALTEN.....	75

**26 BETRIEBSSTUNDEN - ZÄHLER ..... 76**  
**27 INDEX - VERZEICHNIS ..... 78**

# 1 Senden eines Signales auf mehrere Gruppenadressen

Wenn ein Telegramm auch an andere Gruppenadressen gesendet werden soll, kann die untenstehende Schaltung verwendet werden. Jedesmal, wenn ein Telegramm auf das Signal "Eingang1" empfangen wird, werden die Werte der drei Signale "Ausgang1", "Ausgang2" und "Ausgang3" auf den Bus gesendet.

Schaltung:



Der erste TRA-Block kopiert den Wert in die drei Signale. Der zweite TRA-Block kopiert das Eingangs-Kontrollsignal auf die Kontrollsignale der Ausgänge. Dies bewirkt, dass die Ausgangs-Signale immer gesendet werden, auch wenn der Wert des Eingangs-Telegrammes nicht geändert hat.

Anwendungsbeispiel:

Ein zentraler Taster soll alle Lichter in einem Haus löschen. Normalerweise wird diese Funktion mit dem ETS gelöst, indem den Aktoren eine zusätzliche Gruppenadresse zugewiesen wird. Sind aber die Anzahl Gruppenadressen in den Aktoren erschöpft, kann das Problem mit der obigen Schaltung einfach gelöst werden. Das Funktionsmodul empfängt das Telegramm des Tasters und sendet es an die Gruppenadressen aller Licht-Aktoren weiter. Man sieht auch, dass es wichtig ist, dass der zweite TRA-Block die Kontrollsignale kopiert, denn der Taster wird immer ein "Aus" auf den Bus senden. Der Wert des Signales ändert sich also nicht, muss aber trotzdem gesendet werden.

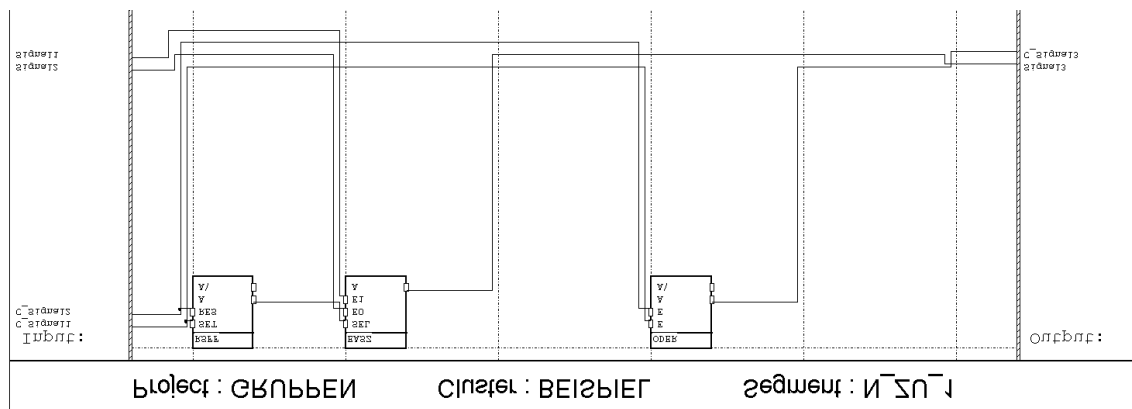
Tips:

- Auch interne Signale können mit einem TRA-Block auf verschiedene Signale aufgeteilt werden. Dann muss allerdings kein Kontrollsignal kopiert werden.

## 2 Steuern eines Aktors mit mehreren Sensoren

Oft muss ein Aktor mehrere Gruppenadressen empfangen, damit er von verschiedenen Tasten, oder anderen Funktionen, gesteuert werden kann. Reichen die vorhandenen Gruppenadressen im Aktor aber nicht aus, kann diese Verknüpfung mit der folgenden Schaltung im Funktionsmodul programmiert werden.

Schaltung:



Mit dieser Grundschaltung können zwei Signale miteinander verknüpft werden. Sie funktioniert folgendermassen: Mit dem EASZ-Block wird entweder der Wert von Signal1 oder von Signal2 auf den Ausgang Signal3 geschaltet. Der RSFF-Block steuert den SEL-Eingang des EASZ-Blockes und schaltet immer das Signal durch, welches zuletzt empfangen wurde. Mit dem ODER-Block werden die beiden Kontrollsignale verknüpft und auf das Kontrollsignal von Signal3 gegeben. Wenn also ein Telegramm auf eines der beiden Eingangs-Signale empfangen wird, sendet das Funktionsmodul immer auch das Signal3 auf den Bus.

Beispiel: Auf Signal1 wird der Wert 0 empfangen. Der Ausgang des RSFF-Blockes wird 1 und der EASZ-Block schaltet die empfangene 0 auf Signal3. Unabhängig davon, ob Signal3 vorher 0 oder 1 war, wird Signal3 gesendet, weil das Kontrollsignal einen Taskzyklus lang auf 1 ist.

Wird nun wieder ein Telegramm auf Signal1 empfangen, schaltet der EASZ-Block immer noch Signal1 durch. Erst wenn dann auf Signal2 ein Telegramm empfangen wird, schaltet der EASZ-Block Signal2 auf das Signal3.

Diese Schaltung kann auch in Serie geschaltet werden, damit mehr als zwei Eingangssignale miteinander verknüpft werden können.

Werden während einem Taskzyklus beide Eingangs-Signale empfangen, wird nur der Wert von Signal2 auf Signal3 gesendet, weil der RES-Eingang des RSFF-Blockes Priorität hat.



## **EIB System**

### **Funktionsmodul und FMTool**

Tips:

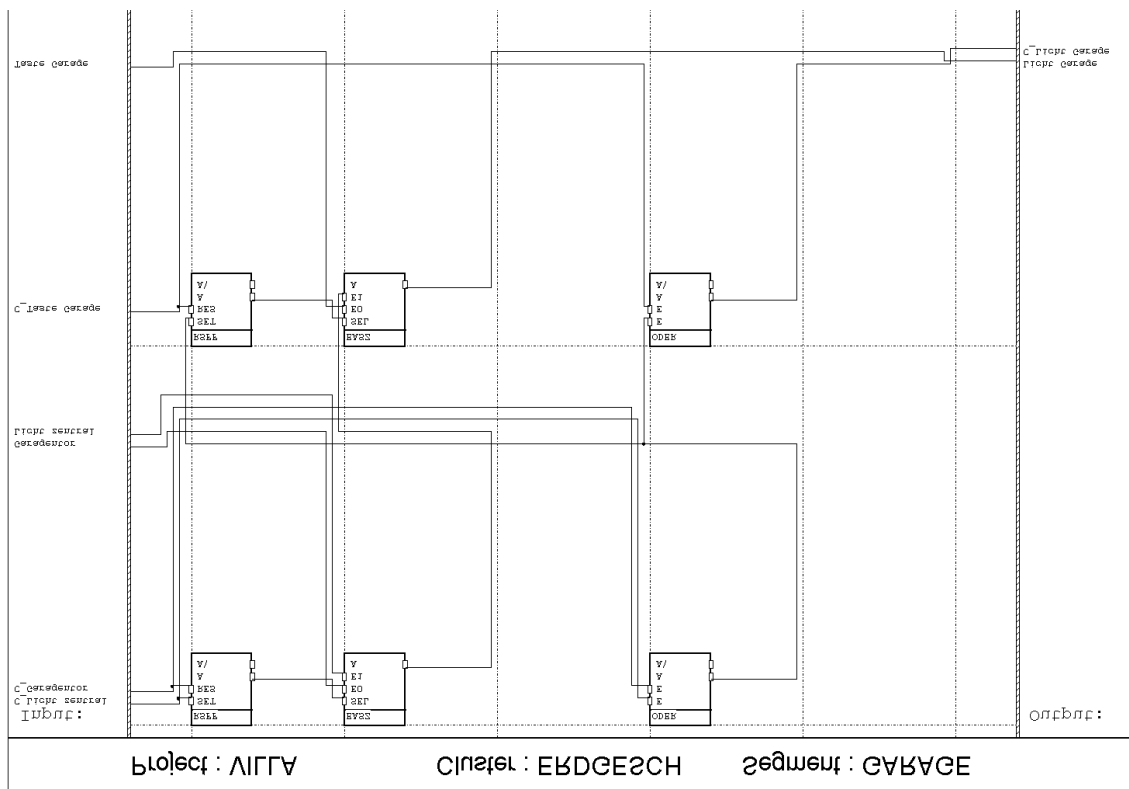
- In dieser Schaltung können die Eingangs- und Ausgangssignale alle möglichen Datentypen haben. Es muss nur der Datentyp des EASZ-Blockes angepasst werden.

**Anwendungsbeispiel:**

Das Licht einer Garage soll von drei verschiedenen Busgeräten ein- und ausgeschaltet werden können. Als erstes ist ein Taster vorhanden, mit dem das Licht von Hand ein- und ausgeschaltet werden kann. Dann gibt es einen Taster mit dem alle Lichter im Haus gelöscht werden können. Das dritte Gerät ist ein Schlüsselschalter ausserhalb der Garage, mit dem das Garagentor aufgemacht werden kann. Dieser Schalter gibt beim Drehen des Schlüssels ein Ein-Telegramm auf den Bus und nach einer bestimmten Zeit wieder ein Aus-Telegramm, um das Tor wieder zu schliessen. Das Licht soll beim Aufgehen des Tors einschalten und beim Zugehen wieder ausschalten.

Wir gehen davon aus, dass unser Licht-Aktor nur eine Gruppenadresse empfangen kann. Es ist also nicht möglich, dieses Problem mit einer direkten Verknüpfung auf dem Bus zu lösen. Darum empfangen wir die Telegramme der drei Schalter mit dem Funktionsmodul, verknüpfen diese und schalten das Licht über eine vierte Gruppenadresse.

**Schaltung:**



In der ersten Zeile werden die beiden Signale "Garagentor" und "Licht zentral" miteinander verknüpft. In der zweiten Zeile werden das Signal "Taste Garage" und das verknüpfte Signal aus der ersten Zeile miteinander verknüpft. Das Ausgangssignal vom ersten ODER-Block wird also in der zweiten Zeile als Kontrollsignal und das Ausgangssignal vom EASZ-Block als Signalwert verwendet.

**EIB System**  
**Funktionsmodul und FMTool**

### 3 Vergleichen von Werten

---

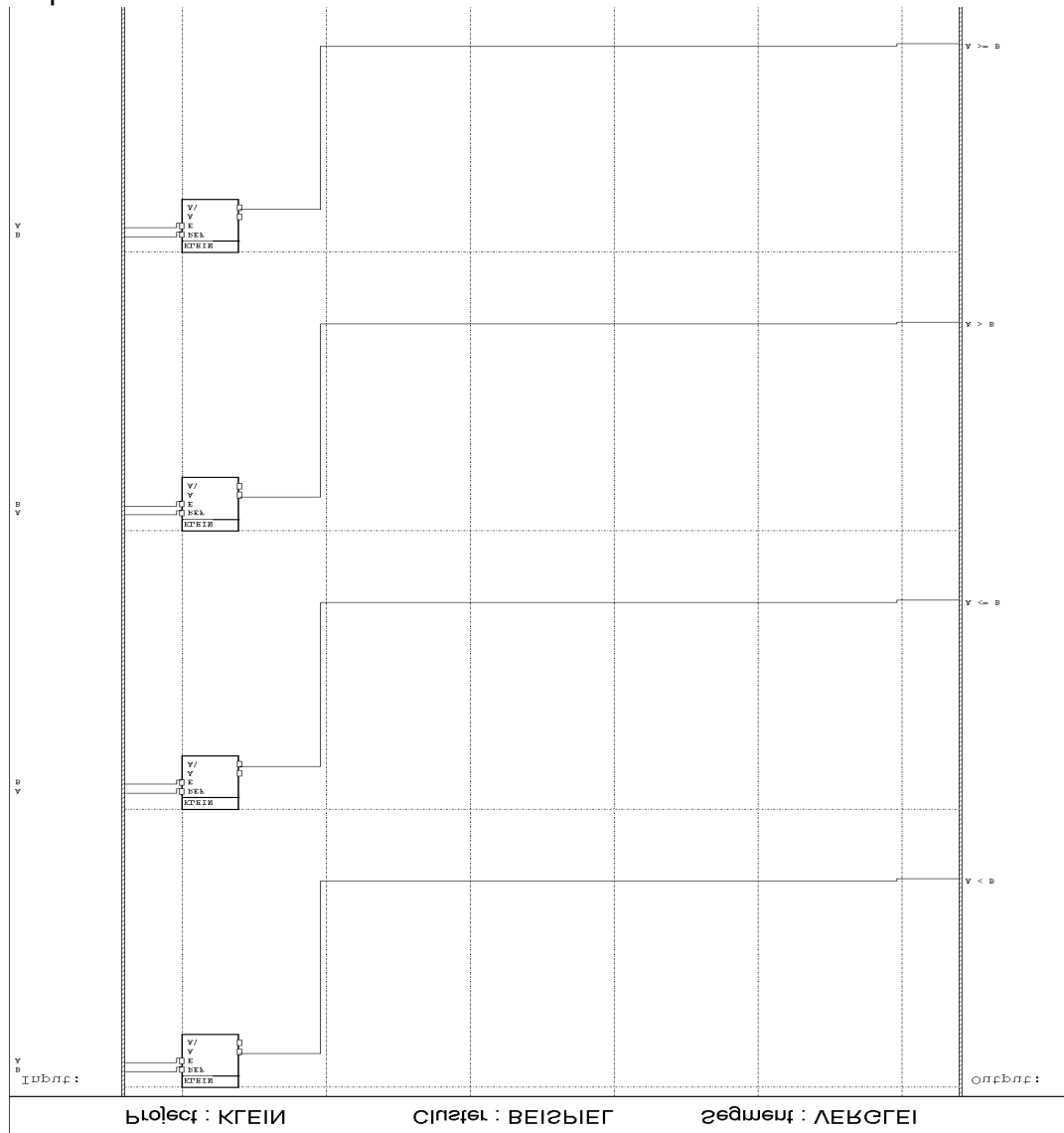
Werte können im Funktionsmodul mit dem KLEIN-Block verglichen werden. Dieser Block kann für folgende Vergleiche verwendet werden: Kleiner als, grösser als, kleiner oder gleich, grösser oder gleich. Es kommt dabei nur darauf an, wie die Signale angeschlossen werden.

Die untenstehende Tabelle zeigt, wie die einzelnen Signale für die verschiedenen Vergleiche anzuschliessen sind. Als zu vergleichende Signale heissen A und B. Der Ausgang wird C genannt und ist 1 wenn der Vergleich wahr ist und 0 wenn er nicht zutrifft.

Vergleich	Signal A	Signal B	Signal C
$C = A < B$	E	REF	A
$C = A \leq B$	REF	E	A\
$C = A > B$	REF	E	A
$C = A \geq B$	E	REF	A\

**EIB System**  
**Funktionsmodul und FMTool**

Beispiele:

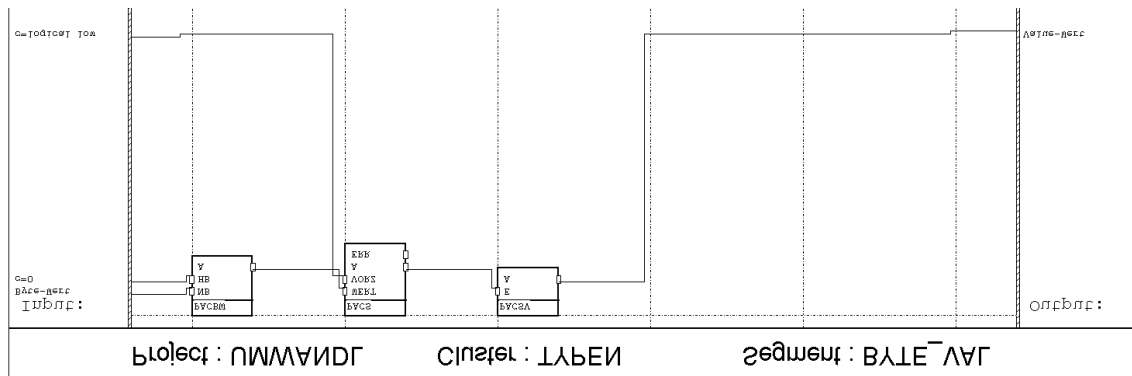


Auf Gleichheit kann mit dem GLEICH-Block geprüft werden.

## 4 Datentyp-Umwandlung von Byte in Value

Oft muss ein Signal vom Datentyp Byte, Word oder SINT in ein Signal vom Datentyp Value umgerechnet werden. Das untenstehende Beispiel zeigt, wie dies gemacht wird. Da die Umwandlung über die Datentypen Word und Sint führt, dient dieses Beispiel auch für die Umwandlung dieser Datentypen.

Schaltung:

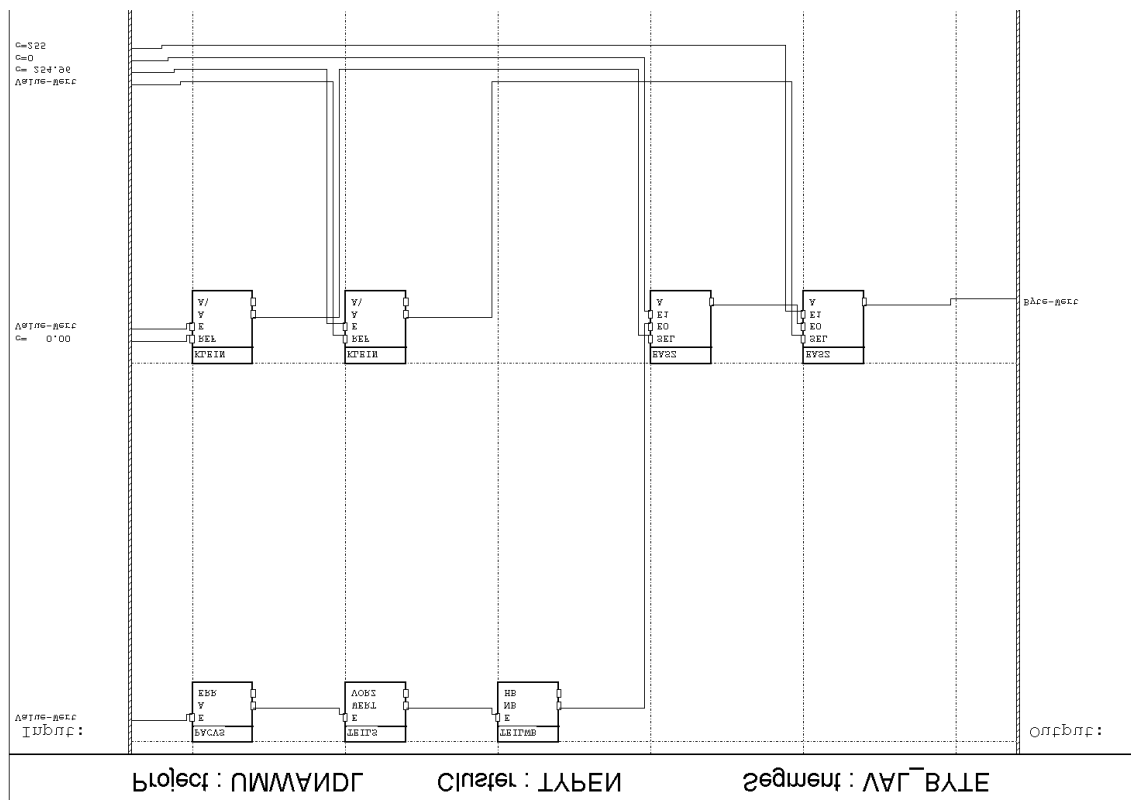


Der PACBW-Block fügt das Byte-Signal in das niederwertigere Byte von einem Word-Signal. Das höherwertige Byte wird auf 0 gesetzt. Das Word-Signal wird mit dem PACS-Block in ein Sint-Signal gewandelt. Das Vorzeichen wird mit der Konstante 0 positiv gemacht (VORZ-Eingang). Da der Wert am WERT-Eingang nicht grösser als 255 werden kann, wird der PACS-Block die Umwandlung in ein Sint-Signal immer vornehmen können (kein Überlauf). Das Sint-Signal wird dann vom PACSV-Block in ein Signal vom Datentyp Value gewandelt. Auch diese Umwandlung ist immer möglich, da der Wertebereich am Eingang vom Datentyp Value immer abgedeckt ist.

## 5 Datentyp-Umwandlung von Value in Byte

Bei der Umwandlung eines Value-Signales in ein Byte-Signal muss berücksichtigt werden, dass die Umwandlung nicht den gesamten Eingangsbereich abdecken kann. Wir müssen definieren was geschieht, wenn der Value-Wert grösser als 255 oder kleiner als 0 ist.

In diesem Beispiel wird bei Werten grösser als 255 am Ausgang immer der Wert 255 ausgegeben und bei negativen Werten wird immer 0 ausgegeben.



Das Value-Signal wird durch den PACVS-Block in ein Sint-Signal, anschliessend mit dem TEILS-Block in ein Word-Signal und schliesslich mit dem TEILWB in ein Byte-Signal gewandelt.

Mit dem ersten KLEIN-Block wird entschieden, ob der Wert vom Value-Signal negativ (kleiner als 0) ist. Ist er negativ, schaltet der erste EASZ-Block die Konstante 0 auf seinen Ausgang. Der zweite KLEIN-Block testet, ob der Value-Wert grösser als 255.96 ist. Trifft dies zu, schaltet der zweite EASZ-Block die Konstante 255 auf den Ausgang.

Ist der Value-Wert also im erlaubten Bereich, wird das Byte-Signal vom TEILWB-Block unverändert über die zwei EASZ-Blöcke auf das Signal "Byte-Wert" durchgeschaltet. Ist er zu klein oder zu gross, wird entweder 0 oder 255 ausgegeben.

Tips:

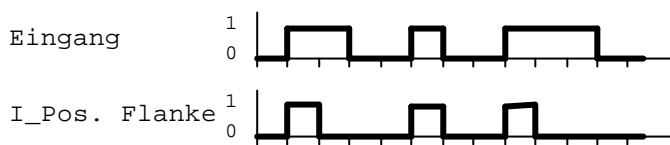
- Falls sichergestellt ist, dass der Wertebereich von Value-Signal immer zwischen 0 und 255 liegt, kann natürlich auf die Schaltung in der zweiten Zeile verzichtet werden.
- Diese Schaltung rundet bei der Umrechnung immer ab (die Kommastellen werden abgeschnitten). Soll der Ausgangswert aber gerundet sein, kann vor dem PACVS-Block ein ADD-Block eingefügt werden, der 0,5 zum Value-Signal dazuzählt.



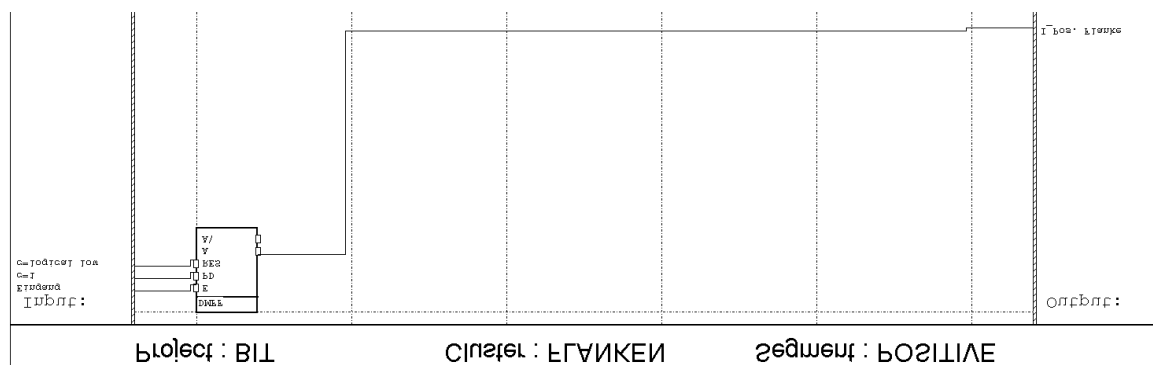
## 6 Positive Flanken erkennen

Oft muss festgestellt werden, wann ein Bit-Signal von 0 auf 1 wechselt. (Dies wird als positive oder steigende Flanke bezeichnet.) Die hier gezeigte Schaltung erzeugt einen Puls auf dem Signal 'I\_Pos. Flanke', wenn am Eingang eine positive Flanke vorkommt. Der Puls dauert einen Taskzyklus lang.

Zeitdiagramm:



Schaltung:

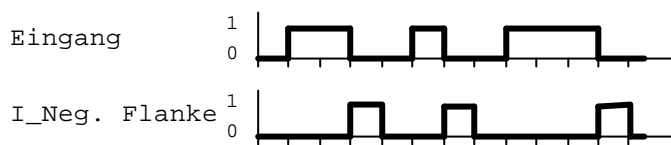


Da der E-Eingang vom DMFF-Block nur auf positive Flanken reagiert, kann er einfach für diesen Zweck verwendet werden.

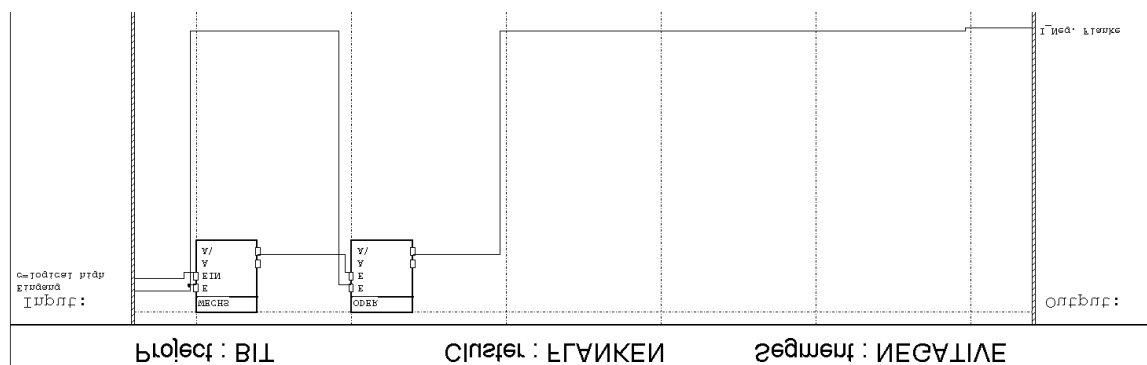
## 7 Negative Flanken erkennen

Vielfach muss festgestellt werden, wann ein Bit-Signal von 1 auf 0 wechselt (was einer negativen oder fallenden Flanke entspricht). Die hier angewendete Schaltung erzeugt einen Puls auf dem Signal 'I\_Neg. Flanke', wenn am Eingang eine negative Flanke vorkommt. Der Puls dauert einen Taskzyklus lang.

Zeitdiagramm:



Schaltung:



Wenn der WECHS-Block eine Änderung am Eingang feststellt (positive oder negative Flanke), hat er kurz eine 0 am invertierten Ausgang (A). Ist der Eingang im selben Augenblick 0 (nur nach einer negativen Flanke), ist der invertierte Ausgang des ODER-Blockes auf 1. Der Ausgang geht also nur bei den negativen Flanken auf 1. Der WECHS-Block hat Datentyp Bit.

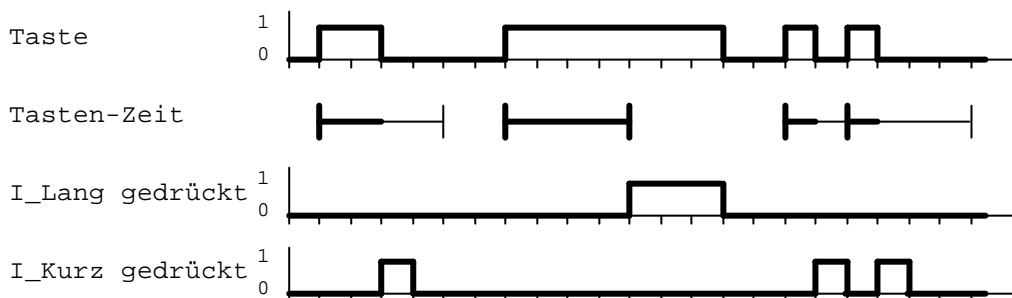
## 8 Erkennen von langen und kurzen Tastendrücken

---

Die vorliegende Schaltung erkennt, ob ein Puls länger oder kürzer als eine bestimmte Zeit dauert. Dies wird mit den beiden Ausgängen 'I\_Lang gedrückt' und 'I\_Kurz gedrückt' angezeigt. 'I\_Kurz gedrückt' ist nur während einem Taskzyklus auf 1. 'I\_Lang gedrückt' ist solange auf 1, wie der Eingangs-Puls länger ist als die vorgegebene Zeit.

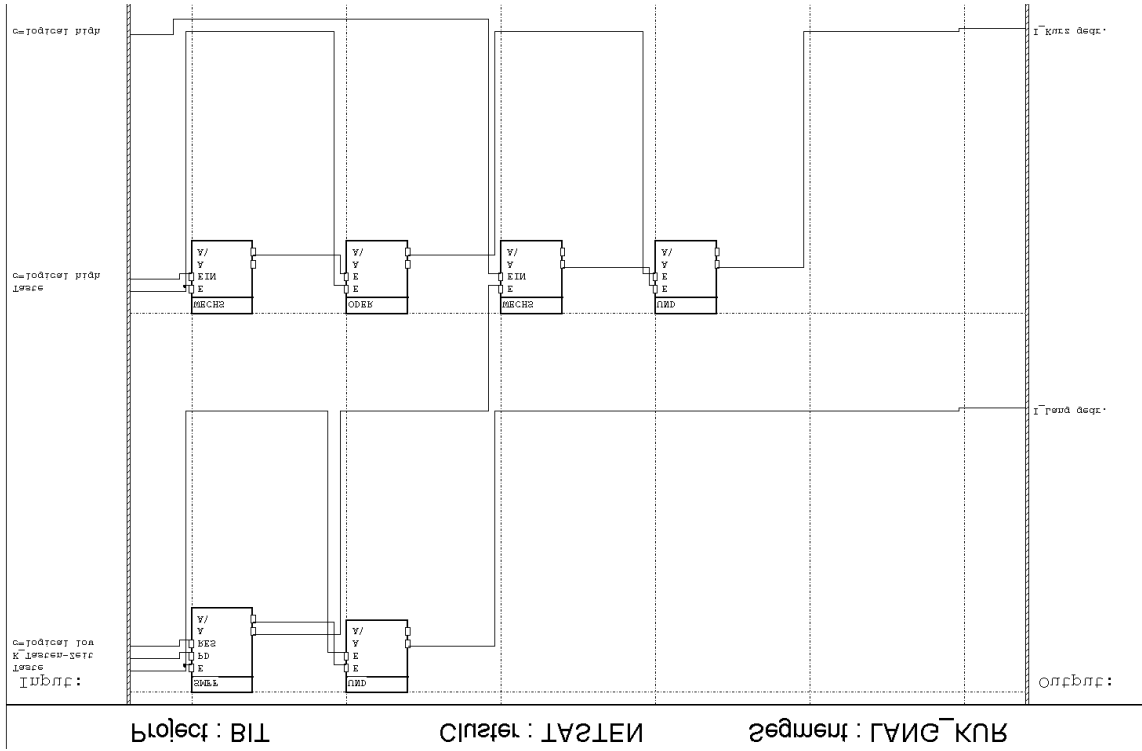
Diese Schaltung wird vorallem zur Auswertung von Tasten verwendet, die zum Dimmen oder Steuern einer Jalousie verwendet werden. Darum die Namen der Ausgänge 'I\_Lang gedrückt' und 'I\_Kurz gedrückt'. Der Taster muss so konfiguriert sein, dass er beim Drücken der Taste ein EIN-Telegramm auf den Bus sendet und beim Loslassen ein AUS-Telegramm sendet.

Zeitdiagramm:



In diesem Beispiel ist der Wert des Signals 'K\_Tasten-Zeit' auf 4 eingestellt.

Schaltung:



## EIB System

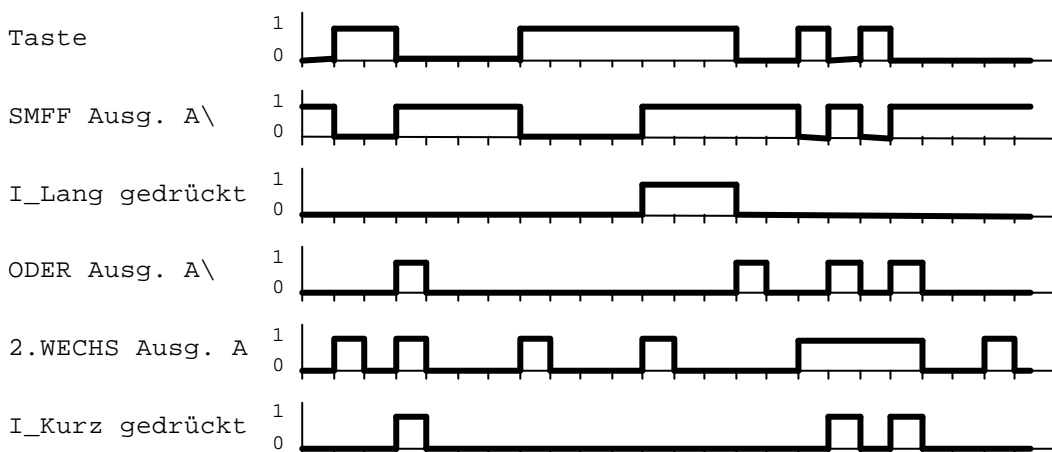
### Funktionsmodul und FMTool

Der SMFF-Block in der ersten Zeile erzeugt einen Referenz-Puls, mit dem der Puls der Taste verglichen wird. Ist der Puls der Taste länger, wird der Ausgang des UND-Blockes 1, bis die Taste wieder losgelassen wird. Dieses Signal zeigt somit einen langen Tastendruck an.

Der erste WECHS-Block und der ODER-Block in der zweiten Zeile detektieren die negativen Flanken am Eingang (siehe auch 'Negative Flanken erkennen'). Der zweite WECHS-Block stellt fest, wann sich der Ausgang des SMFF-Blockes ändert. Fallen nun die Änderung am SMFF-Block und eine negative Flanke am Eingang zusammen (dies kann nur vorkommen wenn die Pulsdauer des SMFF-Blocks durch den Eingangs-Puls noch nicht überschritten wurde), wird dies als kurzer Tastendruck interpretiert. Der UND-Block hat in diesem Fall während einem Taskzyklus eine 1 am Ausgang (Signal 'I\_Kurz gedrückt').

Ist der Eingangs-Puls also weniger oder gleich viele Taskzyklen lang wie die am SMFF-Block eingestellte Pulsdauer (Eingang PD), wird am Ausgang 'I\_Kurz gedrückt' ein Puls erscheinen. (Vergleiche mit detailliertem Zeitdiagramm.)

Detailliertes Zeitdiagramm:



Der Wert des Signals 'K\_Tasten-Zeit' ist auf 4 eingestellt.

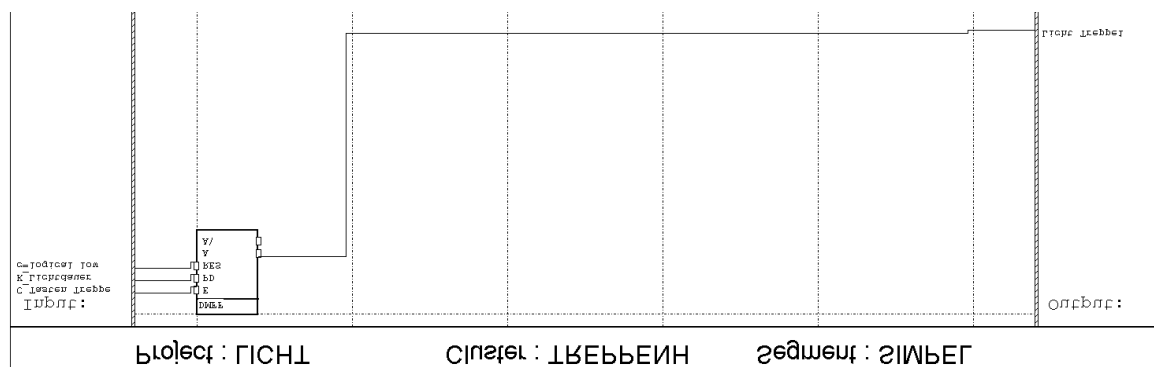
Tips:

- Kann auf das Erkennen eines kurzen Tastendruckes verzichtet werden, kann die untere Zeile der Schaltung weggelassen werden.
- Diese Schaltung kann zum Steuern einer Jalousie verwendet werden.
- Diese Schaltung kann zum Dimmen von Licht verwendet werden.

## 9 Treppenhauslicht, einfach

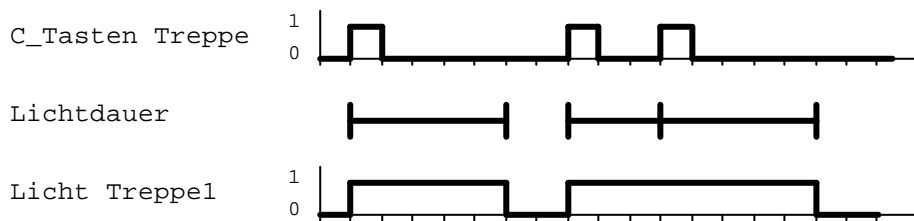
Die einfachste Treppenhaussteuerung kann mit dem DMFF-Block programmiert werden. Wird auf die Taste gedrückt, schaltet das Licht ein. Nach einer bestimmten Zeit schaltet das Licht automatisch wieder aus. Wird innerhalb dieser Zeit nochmals auf die Taste gedrückt, dauert es wieder die voreingestellte Zeit bis das Licht löscht. Die Tasten müssen bei jedem Tastendruck ein Telegramm auf den Bus senden. Es kommt nicht darauf an, ob es ein Ein- oder ein Aus-Telegramm ist. Die Taster können also als normale Umschalter programmiert sein.

Schaltung:



Jedesmal wenn das Kontrollsignal am Eingang des DMFF-Blockes einen Puls hat (Telegramm wurde empfangen), erscheint am Ausgang ein Puls mit der programmierten Pulsdauer. Der Puls wird verlängert, wenn der Ausgang schon 1 ist und erneut ein Telegramm empfangen wird (retriggerbares Flip-Flop).

Zeitdiagramm:



Tips:

- Damit mehrere Tasten das Licht einschalten können, müssen diese auf die selbe Gruppenadresse senden. Haben die Taster aber aus irgend welchen Gründen verschiedene Gruppenadressen, können im Funktionsmodul die Kontrollsignale dieser Gruppenadressen miteinander durch ein ODER-Block verknüpft werden.

## **EIB System**

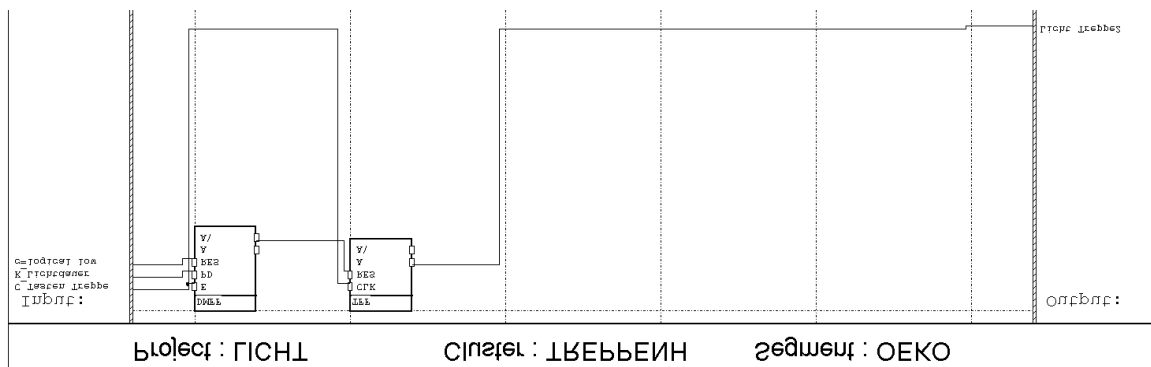
### **Funktionsmodul und FMTool**

- Sollen die LED's der Taster den Zustand des Lichts anzeigen oder als Positionsleuchte im dunkeln Raum dienen, können diese LED's über eine andere Gruppenadresse geschaltet werden. Diese Gruppenadresse muss als zweite Adresse dem Taster-Objekt zugewiesen werden und das Schreiben-Flag muss gesetzt sein.
- Soll die Lichtdauer flexibel einstellbar sein, kann der Wert über den Bus übermittelt werden.
- Diese Schaltung sollte in einem schnelleren Task laufen, damit die Reaktionszeit auf einen Tastendruck nicht zu lang ist (20..100ms)

## 10 Treppenhauslicht, mit Ökoschaltung

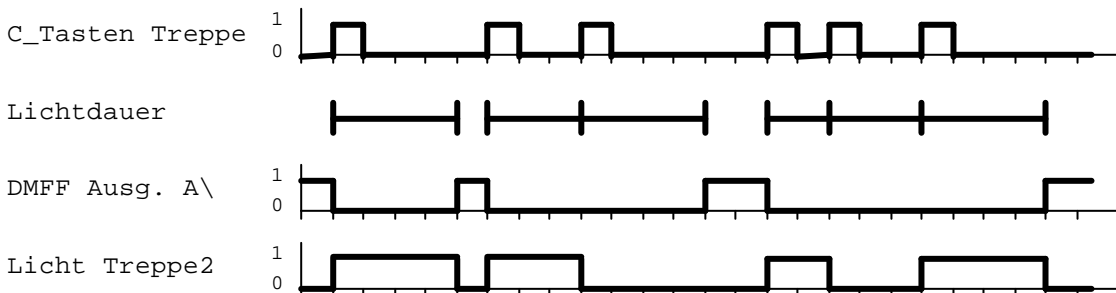
Mit dieser Treppenhaussteuerung kann der Benutzer Strom sparen. Wenn einmal auf eine Taste gedrückt wird, geht das Licht an und löscht nach einer bestimmten Zeit wieder (wie bei der einfachen Treppenhaussteuerung). Man hat aber die Möglichkeit, durch erneutes Drücken auf eine Taste, das Licht wieder zu löschen. Die Tasten müssen bei jedem Tastendruck ein Telegramm auf den Bus senden. Es kommt nicht darauf an, ob es ein Ein- oder ein Aus-Telegramm ist. Die Taster können also als normale Umschalter programmiert sein.

Schaltung:



Jedesmal wenn ein Telegramm der Tasten empfangen wird, schaltet der TFF-Block um von 0 auf 1 oder umgekehrt. Dies funktioniert also wie ein gewöhnlicher Umschalter. Doch wird jedesmal auch der DMFF-Block gestartet, der nach der programmierten Zeit (Lichtdauer) den TFF-Block wieder auf 0 setzt. (Vergleiche mit dem Zeitdiagramm.)

Zeitdiagramm:



Tips:

- Hier gelten die gleichen Tips wie für das einfache Treppenhauslicht.



## 11 Treppenhauslicht, mit Dauerlichtschaltung

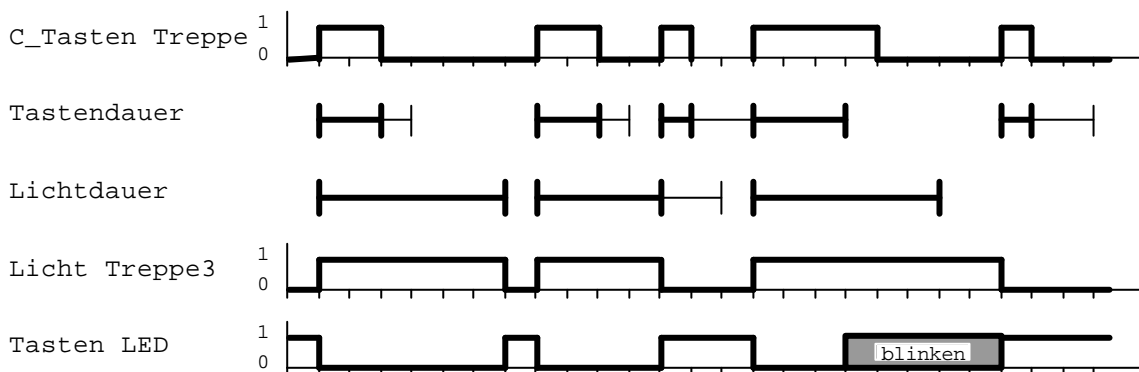
Diese Schaltung arbeitet wie das Treppenhauslicht mit Ökoschaltung: Das Licht kann mit einem Tastendruck eingeschaltet werden und löscht nach einer programmierten Zeit wieder, wenn nicht vorher mit einem erneuten Tastendruck das Licht ausgemacht wurde. Zusätzlich kann das Licht aber durch langes Drücken auf eine Taste dauernd eingeschaltet werden. Es löscht dann erst wieder, wenn nochmals auf eine Taste gedrückt wird.

Damit der Benutzer darauf aufmerksam gemacht wird, dass das Licht im "Dauerbetrieb" brennt, blinken die LED's der Taster. So wird das Löschen des Lichts nicht vergessen. Ansonsten sind die LED's dunkel wenn das Licht brennt und leuchten, wenn es nicht brennt (Positions-Anzeige).

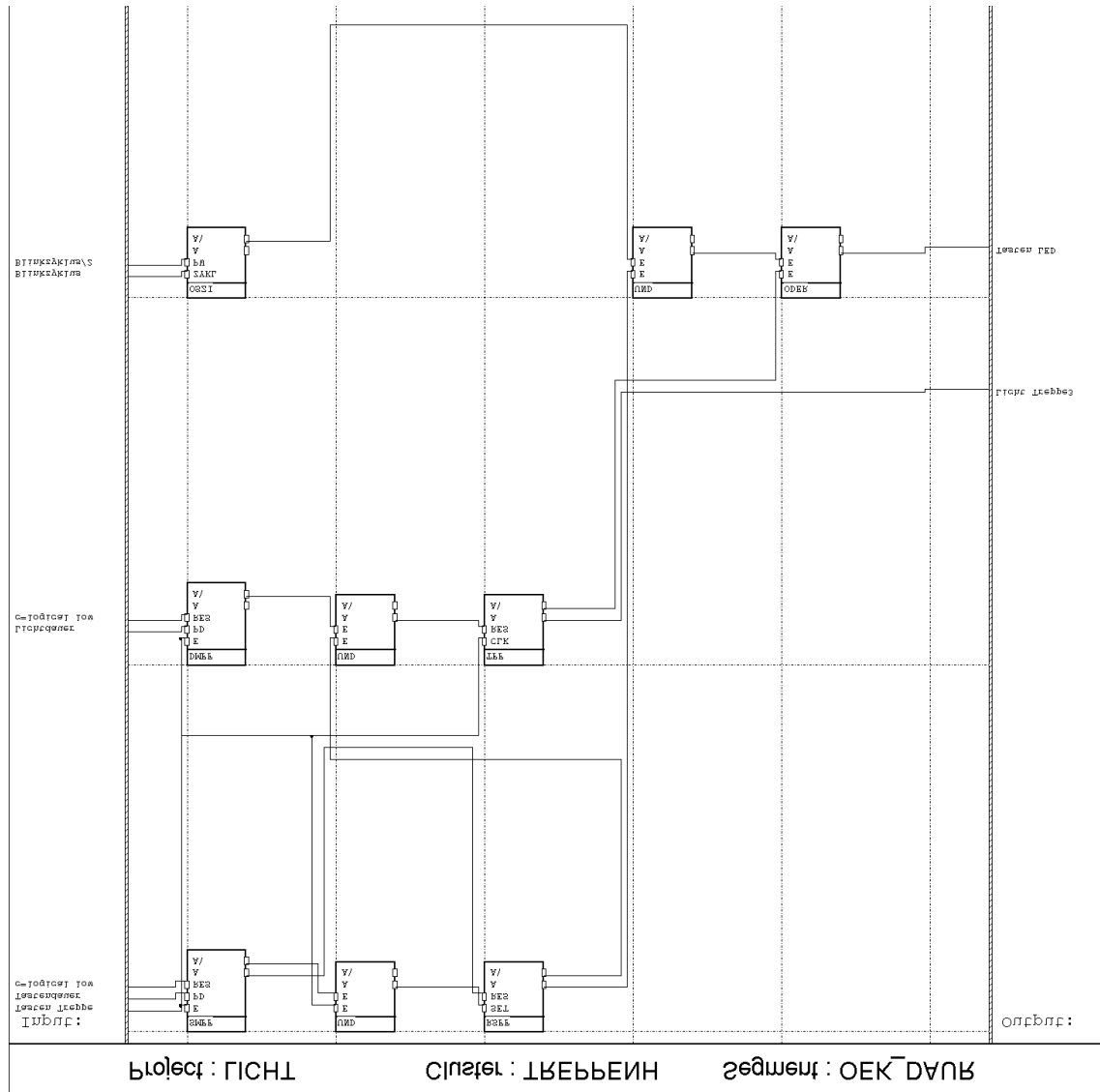
Diese Schaltung kann z.B. in einer Garage verwendet werden die auch als Werkstatt dient. Wird nur das Auto parkiert oder damit weggefahren soll das Licht automatisch löschen. Wird in der Garage gearbeitet, soll das Licht nicht selber löschen und kann also in den Dauerbetrieb geschaltet werden.

Die Tasten müssen für diese Schaltung so konfiguriert werden, dass sie beim Drücken ein Ein-Telegramm und beim Loslassen ein Aus-Telegramm senden. Nur so kann das Funktionsmodul feststellen, wenn die Taste lange gedrückt wird. Die LED's der Taster müssen über eine eigene Gruppenadresse angesteuert werden können..

Zeitdiagramm:



Schaltung:



In der ersten Zeile wird überprüft, ob die Taste lang gedrückt wurde oder nur kurz. Der RSFF-Block speichert, ob die Schaltung im "Dauerlicht-Modus" ist oder nicht. Ist der Ausgang A auf 1, ist der Dauerlicht-Modus aktiv. Der SMFF-Block und der UND-Block stellen fest, wenn die Taste länger gedrückt ist als die Tastendauer (siehe "Erkennen von langen und kurzen Tastendrücken") und setzen in diesem Fall den RSFF-Block auf 1. Durch den Ausgang A des SMFF-Blockes wird der RSFF-Block bei jedem Tastendruck auf 0 gesetzt.. Da der Reset-Eingang des RSFF-Blockes Vorrang hat, wird der Dauerlicht-Modus nur aktiv, wenn die Taste lange gedrückt wird. Er wird aber bei jedem weiteren Tastendruck wieder inaktiv.

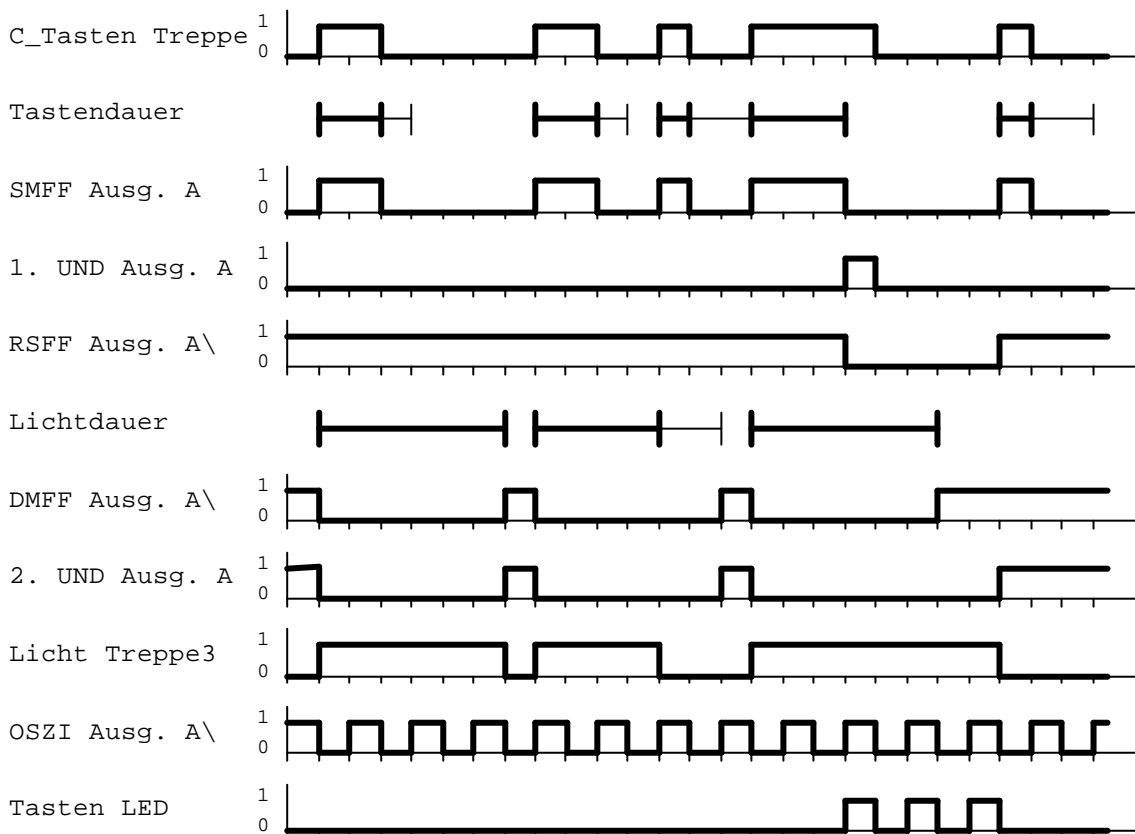
## **EIB System**

### **Funktionsmodul und FMTool**

Die zweite Zeile enthält im wesentlichen die Schaltung vom Treppenhauslicht mit Ökoschaltung (DMFF- und TFF-Block). Jedoch kann der TFF-Block jetzt nur noch ein Reset erhalten, wenn der Dauerlicht-Modus inaktiv ist (Ausgang A\ des RSFF-Blockes). Der UND-Block verhindert also das Abschalten des Lichtes nach Ablauf der Lichtdauer im Dauerlicht-Modus.

In der dritten Zeile werden die Tasten-LED's angesteuert. Der OSZI-Block erzeugt den Blinktakt für die LED's. Der ODER-Block schaltet die LED's immer ein, wenn das Licht nicht brennt (Ausgang A\ des TFF-Blockes). Andernfalls schaltet der UND-Block das Oszillator-Signal auf den anderen Eingang des ODER-Blockes, wenn der Dauerlicht-Modus aktiv ist. Dann blinken die LED's. (Vergleiche mit detailliertem Zeitdiagramm.)

Detailliertes Zeitdiagramm:



Tips:

- Damit mehrere Tasten das Licht einschalten können, müssen diese auf die selbe Gruppenadresse senden. Haben die Taster aber aus irgend welchen Gründen verschiedene Gruppenadressen, können im Funktionsmodul die Signale dieser Gruppenadressen durch ein ODER-Block miteinander verknüpft werden.
- Soll die Lichtdauer flexibel einstellbar sein, kann der Wert über den Bus übermittelt werden.
- Diese Schaltung sollte in einem schnellen Task laufen, damit die Reaktionszeit auf einen Tastendruck nicht zu lang ist (20..100ms).

## **EIB System**

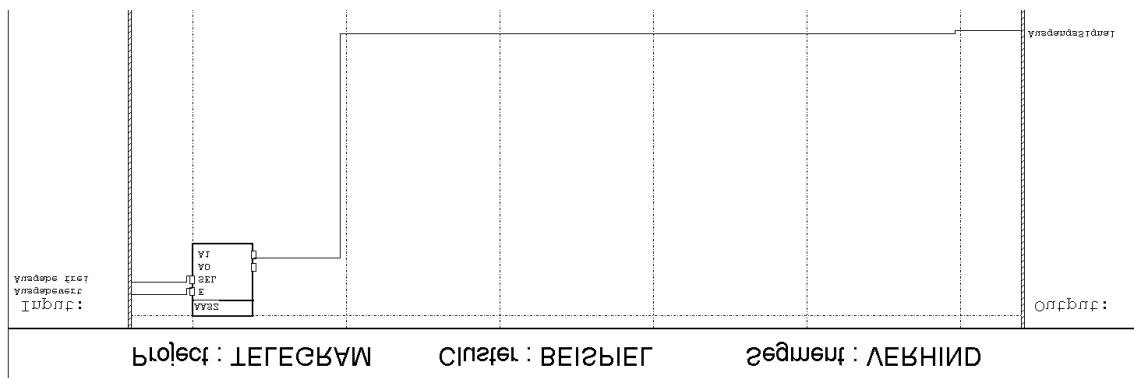
### **Funktionsmodul und FMTool**

- Die Dauer für einen langen Tastendruck ("Tastendauer") soll im Bereich 0.5 bis 1 Sekunden liegen.
- Die LED's sollen nicht zu schnell blinken, damit der Bus und der Ausgangspuffer des Funktionsmodules nicht zu stark belastet werden. Die Konstante "Blinkzyklus" soll ca. 2 Sekunden sein.

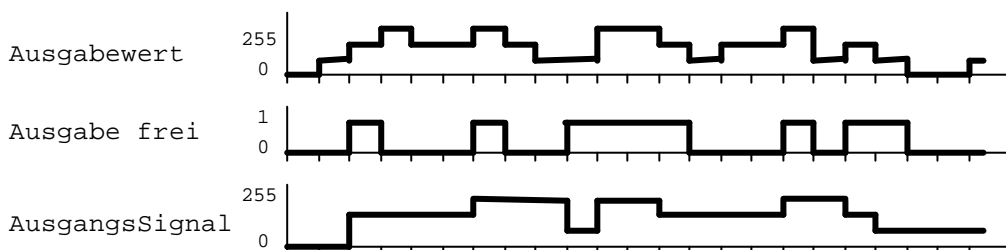
## 12 Ausgabe von Telegrammen verhindern

Wenn sich der Wert eines Ausgangs-Signales ändert, wird normalerweise immer ein Telegramm mit dem neuen Wert gesendet. Oft muss dies aber verhindert werden. Es muss also verhindert werden, dass sich der Signalwert am Ausgang ändert. Dies kann mit dem AASZ-Block erreicht werden.

Schaltung:



Zeitdiagramm:



Das Ausgangssignal wird nur gesendet, wenn das Signal 'Ausgabe frei' 1 ist und sich sein Wert ändert.

Tips:

- Siehe auch das Beispiel 'Zyklische Ausgabe von Werten'.

## 13 Redundanz mit zwei Funktionsmodulen

---

Um die Ausfallsicherheit des Funktionsmodules zu erhöhen, können zwei Funktionsmodule eingesetzt werden, die sich gegenseitig in Ihrer Funktion ablösen können. Je mehr Funktionen das Funktionsmodul in einem Gebäude übernimmt und je wichtiger diese Funktionen sind, desto verheerender wirkt sich ein Ausfall des Funktionsmodules oder der umgebenden Komponenten aus (bei Stromausfall, Leitungsunterbruch, Ausfall von Linienkoppler, Unterbruch durch das Laden einer neuen Applikation usw.)

Das vorliegende Redundanz-Konzept kann bei den meisten dieser Ausfälle die Funktionen ohne grossen Unterbruch fortsetzen.

### Konzept

Zwei Funktionsmodule sind genau gleich programmiert, damit beide alle Funktionen ausführen können. Beide Funktionsmodule empfangen alle Telegramme und berechnen die Ausgangswerte. Es ist aber nur ein Funktionsmodul aktiv und sendet seine Ausgangswerte auf den Bus. Das andere bleibt passiv und sendet keine Telegramme. Fällt nun das aktive Funktionsmodul aus, gibt es seine Ausgänge frei und sendet seine Werte auf den Bus.

Damit die Funktionsmodule feststellen können, ob ihr Partner noch funktioniert, müssen sie sich gegenseitig überwachen. Dies geschieht mit zyklischen Telegrammen (sogenannte Lebenszeichen), die dem anderen Funktionsmodul immer wieder bestätigen, dass es noch lebt. Erhält nun das passive Funktionsmodul keine Lebenszeichen mehr, wird es aktiv und stellt so das Weiterfunktionieren des Systems sicher.

Es müssen also zwei verschiedene Gruppenadressen für die Lebenszeichen der beiden Funktionsmodule vergeben werden.

### Erzeugen der Lebenszeichen

Grundsätzlich soll das Lebenszeichen zyklisch und in gleichmässigen Abständen gesendet werden. Das Lebenszeichen ist ein Telegramm vom Datentyp Bit (EIS1). Zum einen teilt es dem anderen Funktionsmodul mit dass es lebt, zum andern, ob es aktiv oder passiv ist. Der Wert 0 bedeutet passiv, 1 bedeutet aktiv.

Nach einem Reset soll aber das Lebenszeichen nicht sofort gesendet werden. Das Funktionsmodul muss zuerst abwarten ob es ein Lebenszeichen seines Partners empfängt und entscheidet dann, ob es aktiv oder passiv sein muss. (siehe weiter unten). Es gibt auch Situationen (fehlerhafte Hardware oder Applikation) wo das Funktionsmodul nach dem Aufstarten gleich wieder ein Reset auslöst. Würde dann das Lebenszeichen ganz zu Beginn gesendet, würde das andere Funktionsmodul die Fehlfunktion nicht erkennen.

**EIB System**  
**Funktionsmodul und FMTool**

Damit bei einem gleichzeitigen Start ein vorausbestimmtes Funktionsmodul aktiv wird, ist die Wartezeit beim anderen Funktionsmodul etwas länger. Es gibt also ein Funktionsmodul mit Priorität (FM1) und eines ohne Priorität (FM2).

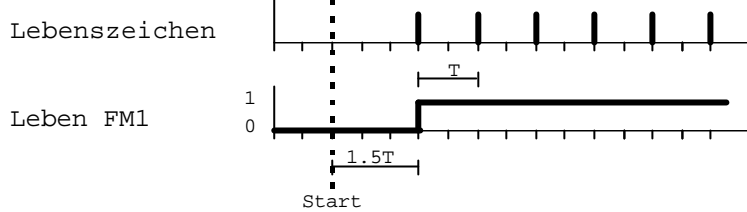


## EIB System

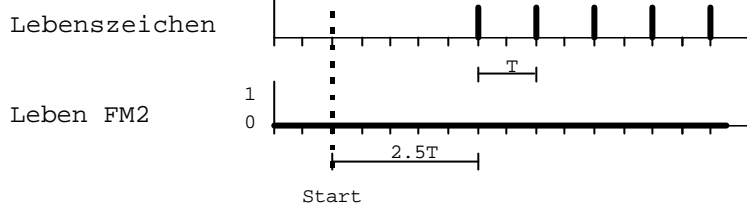
### Funktionsmodul und FMTool

#### Zeitdiagramm der Lebenszeichen:

FM1 mit Priorität:



FM2 ohne Priorität:



#### Legende:

T Zykluszeit der Lebenszeichen

1.5T Wartezeit beim Aufstarten des Funktionsmodules mit Priorität

2.5T Wartezeit beim Aufstarten des Funktionsmodules ohne Priorität

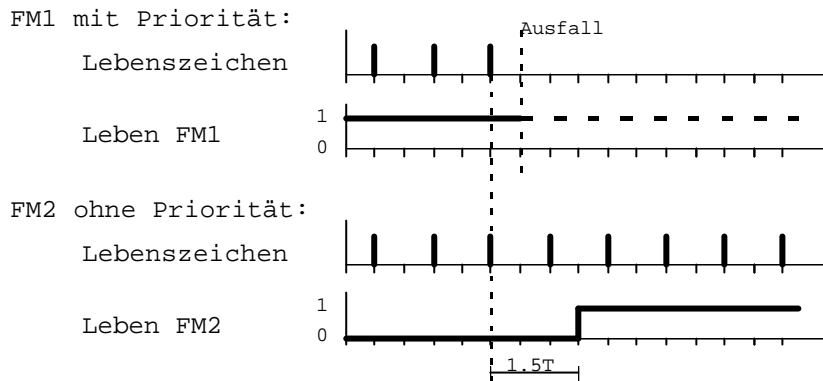
#### Überwachung der Lebenszeichen

Jedesmal, wenn ein Funktionsmodul ein Lebenszeichen seines Partners empfängt, setzt es seine Überwachungszeit neu. In dieser Zeit muss das nächste Lebenszeichen empfangen werden. Wird kein Lebenszeichen empfangen, wird das Funktionsmodul aktiv. Die Überwachungszeit ist so gewählt, dass das Lebenszeichen auch bei einer Verzögerung (überlasteter Bus oder überfüllter Ausgangspuffer des Partner-Funktionsmodules) noch empfangen wird (empfohlene Zeit: 1.5 mal Zykluszeit der Lebenszeichen).

In der Startphase muss mit dem Entscheid, ob aktiv oder passiv, gewartet werden, bis ein Lebenszeichen des Partners empfangen wird. Erst nach dieser Startphase tritt die normale Überwachung in Kraft. Die Startphase ist bei den beiden Funktionsmodulen FM1 und FM2 wiederum unterschiedlich. Das FM2 wartet länger, um dem FM1 bei einem gleichzeitigen Start den Vorrang zu geben.

Normalerweise wird ein Funktionsmodul, das einmal aktiv ist, nie mehr passiv (ausser es wird frisch gestartet). Es gibt aber Fälle die dazu führen könnten, dass dann plötzlich beide Funktionsmodule aktiv sind. Damit dies nicht geschieht, ist das FM2 so programmiert, dass es passiv wird, falls das FM1 ein Lebenszeichen mit dem Wert 1 (=aktiv) sendet.

Beispiel:



Legende:

T Zykluszeit der Lebenszeichen

1.5T Überwachungszeit der Lebenszeichen vom Partner-Funktionsmodul

Fehlermeldung bei Ausfall

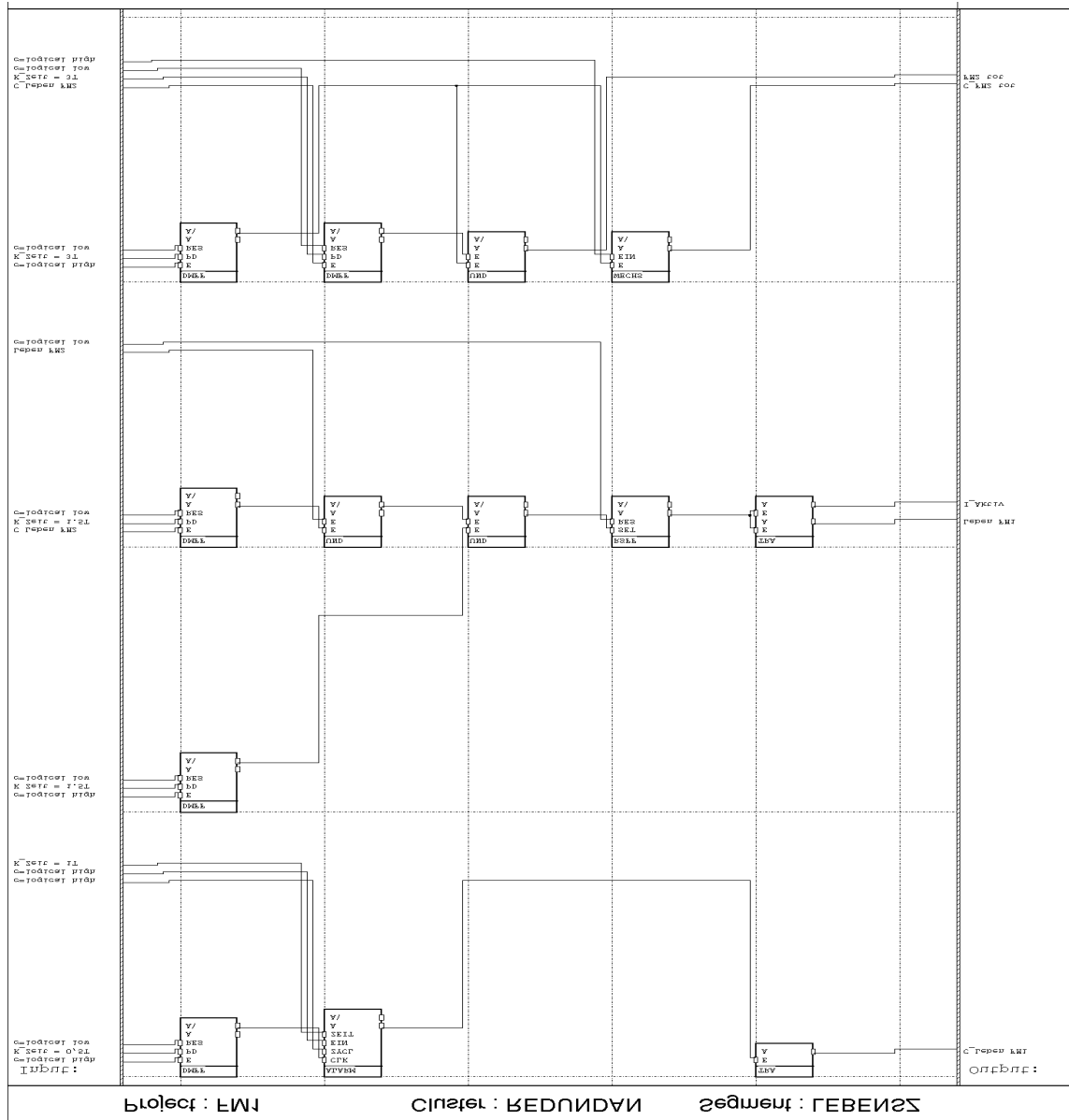
Die Funktionsmodule müssen den Ausfall ihres Partners an eine zentrale Visualisierung oder an Alarmleuchten weiterleiten, damit die Störung erkannt und wieder behoben werden kann. Dafür hat jedes Funktionsmodul eine Gruppenadresse vom Typ Bit, womit es mitteilen kann, ob sein Partner noch funktioniert. Fällt der Partner aus, sendet das 'überlebende' Funktionsmodul ein 1-Telegramm an die Visualisierung. Ist der Fehler wieder behoben, wird ein 0-Telegramm gesendet. In diesem Beispiel wurde die Überwachungszeit der Lebenszyklen für die Fehlermeldung relativ lang gewählt (3 mal die Taskzykluszeit der Lebenszeichen). Dadurch wird bei einem kurzen Unterbruch noch keine Fehlermeldung gesendet. Nach dem Aufstarten wird der Fehlerzustand immer gesendet, damit die Visualisierung sicher den richtigen Wert hat.

# EIB System

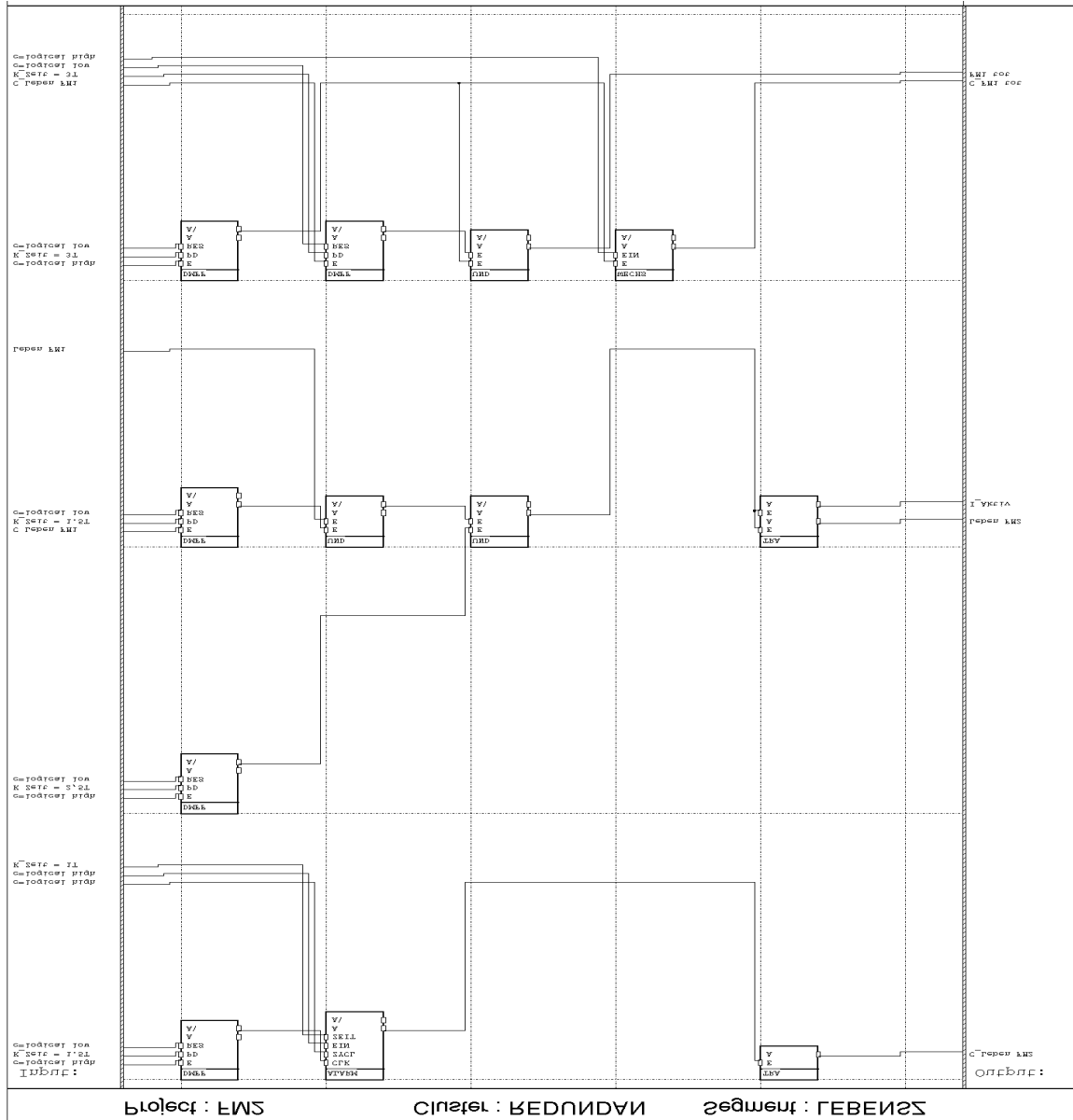
## Funktionsmodul und FMTool

### Schaltungen

#### FM1 mit Priorität:



FM2 ohne Priorität:



Erzeugen der Lebenszeichen:

In der ersten Zeile der beiden Segmente wird das Lebenszeichen zyklisch gesendet. Der DMFF-Block legt die Startverzögerung fest (0,5T für FM1, 1.5T für FM2). Da am E-Eingang dieses Blockes die Konstante 1 anliegt, entspricht dies beim Aufstarten einer positiven Flanke. Dadurch ist der invertierte Ausgang A\ 0, bis die Verzögerungszeit abgelaufen ist. Danach geht der invertierte Ausgang A\ auf 1 und der ALARM-Block wird gestartet (CLK-Eingang). Nach Ablauf der Zykluszeit der Lebenszeichen (1T) wird ein Puls auf das Kontrollsignal von 'C\_Leben FM1' bzw.

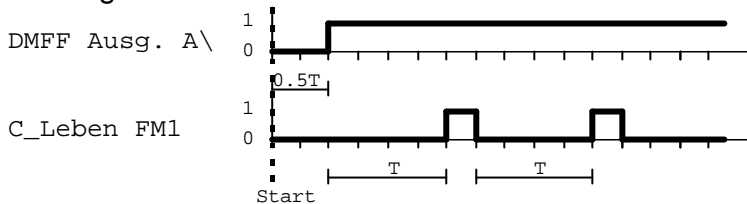
## EIB System

### Funktionsmodul und FMTool

'C\_Leben FM2' gegeben und damit ein Lebenszeichen-Telegramm gesendet. Da der ZYKL-Eingang des ALARM-Blockes auf 1 ist, wird das Lebenszeichen zyklisch immer wieder gesendet.

Der TRA-Block ist nötig, da in der FBL-Version 1.04 oder älter, der ALARM-Block nicht direkt verwendet werden kann, um zyklisch auf ein Kontrollsignal zu schreiben.

Zeitdiagramm:



Man sieht, dass die Zykluszeit der Lebenszeichen in Wirklichkeit um einen Taskzyklus länger ist, als am ZEIT-Eingang des ALARM-Blockes angegeben. Dies kann aber bei Zeit-Werten von einigen Hundert, vernachlässigt werden.

#### Überwachung der Lebenszeichen

Der DMFF-Block in der dritten Zeile überwacht das Lebenszeichen des anderen Funktionsmodules. Immer wenn das Lebenszeichen empfangen wird, wird der Ausgang A des DMFF-Blockes für  $1.5T$  lang auf 1 gesetzt. Kommt das Lebenszeichen immer zur rechten Zeit (im Abstand von  $T$ ), bleibt der Ausgang also immer auf 1. Der erste UND-Block verknüpft dieses Signal mit dem Wert des Lebenszeichens. Ist das andere Funktionsmodul passiv (das Lebenszeichen ist 0) oder das Lebenszeichen wird nicht empfangen, ist der invertierte Ausgang A 1. In diesem Fall muss das Funktionsmodul aktiv werden, ausser es hat erst gerade gestartet.

Für die Startphase ist der DMFF-Block in der zweiten Zeile und der zweite UND-Block verantwortlich. Nach dem Aufstarten ist der invertierte Ausgang A\ des DMFF-Blockes während  $1.5T$  bzw.  $2.5T$  auf 0 und sorgt dafür, dass der Ausgang A des zweiten UND-Blockes nicht 1 wird.

Beim FM2 geht der Ausgang A des zweiten UND-Blockes direkt auf das Signal 'Leben FM1' und auf das interne Signal 'I\_Aktiv', dadurch kann das FM2 wechseln zwischen aktiv und passiv. Beim FM1 geht dieser Ausgang auf ein RSFF-Block. Dieser kann nur einmal gesetzt, aber nie mehr zurückgesetzt werden, da der RES-Eingang an die Konstante 0 angeschlossen ist. Durch diesen Mechanismus kann verhindert werden, dass die Funktionsmodule durch einen dummen Zufall dauernd aktiv und wieder passiv werden.

#### Fehlermeldung bei Ausfall

In der vierten Zeile wird die Fehlermeldung an eine Visualisierung generiert. Der zweite DMFF-Block überwacht das Lebenszeichen des Partner-Funktionsmodules. Falls das Lebenszeichen immer kommt, ist sein invertierter Ausgang immer 0, wenn das andere Funktionsmodul ausfällt 1. Während der Startphase ( $3T$ ) ist der invertierte Ausgang A\ des ersten DMFF-Blockes 0. Dadurch ist während dieser Zeit der

Ausgang des UND-Blockes auch 0 und es wird noch kein Fehlertelegramm 'FM1 tot' bzw. 'FM2 tot' gesendet.

Nach Ablauf der Startphase (wenn der Ausgang A\ vom ersten DMFF-Block von 0 auf 1 geht) erzeugt der WECHS-Block einen Puls auf das Kontrollsignal 'C\_FM1 tot' bzw. 'C\_FM2 tot'. Die Visualisierung erhält so den aktuellen Wert.

#### Steuern der Ausgänge

Wenn das Funktionsmodul passiv ist, müssen die Ausgänge aller Funktionen gesperrt werden. D.h. die Ausgangswerte dürfen sich nicht ändern. Dies kann mit einem AASZ-Block programmiert werden (siehe Beispiel "Ausgabe von Telegrammen verhindern").

Wenn ein Funktionsmodul aktiv wird, sollen alle Ausgangssignale des Funktionsmoduls auf den Bus gesendet werden. Damit wird gewährleistet, dass alle Geräte den gleichen Zustand haben wie die Ausgänge des nun aktiven Funktionsmoduls. Dabei werden die Ausgänge in zwei Gruppen aufgeteilt:

- Ausgänge bei denen nur eine Änderung des Signalwertes eine Funktion auslöst: Bei diesen Signalen wird normalerweise das Kontrollsignal nicht verwendet. Beispiele sind: ein/ausschalten von Licht, dimmen mit Absolutwert, senden von Werten (Helligkeitswert, Ventilstellwert usw.).  
Bei diesen Ausgängen muss der Wert gesendet werden, wenn das Funktionsmodul aktiv wird (siehe Schaltung).
- Ausgänge bei denen auch das Senden des gleichen Signalwertes eine Funktion auslöst: Bei diesen Signalen wird das Kontrollsignal verwendet. Beispiele sind: Relatives Dimmen (heller/dunkler), Jalousie auf/ab, Lamellen auf/ab, usw.  
Bei diesen Ausgängen sollen keine Telegramme gesendet werden, wenn das Funktionsmodul aktiv wird (siehe Schaltung). Es könnten sonst unerwünschte Reaktionen von Jalousien usw. entstehen.

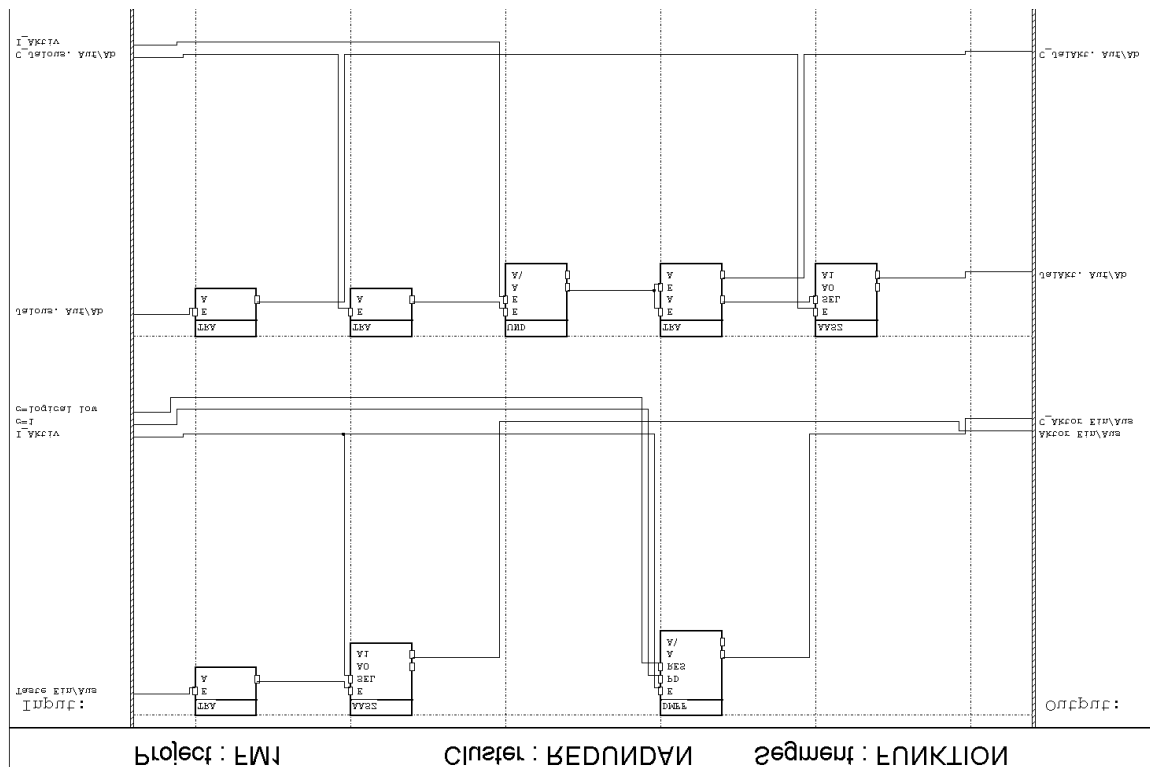
#### Schaltung:

In der folgenden Schaltung ist je ein Beispiel für die zwei Gruppen von Signalen. In der ersten Zeile ist ein Beispiel für die erste Gruppe. Als einfache Funktion soll hier der Wert einer Taste auf einen Aktor gesendet werden, der eine andere Gruppenadresse hat. Der TRA-Block gehört also nicht zur eigentlichen Schaltung um die Ausgänge zu sperren. Er steht nur anstelle von irgend einer Funktion, die ein Ein/Aus-Telegramm erzeugen soll.

Auch in der zweiten Zeile sind die ersten beiden TRA-Blöcke anstelle von irgend einer Schaltung, die z.B eine Jalousie auf und ab bewegen soll. Der erste TRA-Block liefert den Wert für das Ausgangssignal und der zweite TRA-Block liefert das Signal für das dazugehörige Ausgangskontrollsignal.

## EIB System

### Funktionsmodul und FMTool



Die Schaltung ist für beide Funktionsmodule exakt die selbe.

In der ersten Zeile ist ein Beispiel für die erste Gruppen von Signalen (Ausgänge bei denen nur eine Änderung des Signalwertes eine Funktion auslöst). In einem nicht redundanten System würde einfach der Ausgang des TRA-Blockes auf das Ausgangssignal 'Aktor Ein/Aus' geführt. Hier muss aber der Ausgang des TRA-Blockes zuerst auf den AASZ-Block geführt werden. Der AASZ-Block schaltet diesen Ausgang nur auf 'Aktor Ein/Aus' wenn 'I\_Aktiv' 1 ist. Ist das Funktionsmodul passiv ('I\_Aktiv' ist 0) bleibt der Ausgang A1 des AASZ-Blockes unverändert und es wird kein Telegramm generiert. Der DMFF-Block sorgt dafür, dass ein Telegramm gesendet wird, wenn das Funktionsmodul aktiv wird (Er gibt einen Puls auf 'C\_Aktor Ein/Aus' wenn 'I\_Aktiv' von 0 auf 1 wechselt).

In der zweiten Zeile ist ein Beispiel für Signale der zweiten Gruppe (Ausgänge bei denen auch das Senden des gleichen Signalwertes eine Funktion auslöst). Hier wird ein Jalousie-Aktor (Auf/Ab) gesteuert. In einem nicht redundanten System würden die Ausgänge der ersten beiden TRA-Blöcke direkt auf die Ausgangssignale 'JalAkt. Auf/Ab' und 'C\_JalAkt. Auf/Ab' geführt. In einem redundanten System muss das Signal des ersten TRA-Blockes (Daten-Signal) auf einen AASZ-Block geführt werden. Der Ausgang des UND-Blockes ist nur 1, wenn das Signal gesendet werden soll (Ausgang des zweiten TRA-Blockes ist 1) und wenn 'I\_Aktiv' 1 ist. In diesem Fall soll der AASZ-Block das Daten-Signal durchschalten und das Kontrollsignal soll 1 sein. Dafür ist der dritte TRA-Block gebraucht. Das Signal wird also nur gesendet, wenn das Funktionsmodul aktiv ist und es durch einen Steuerbefehl gesendet werden muss.

**Viele Ausgänge**

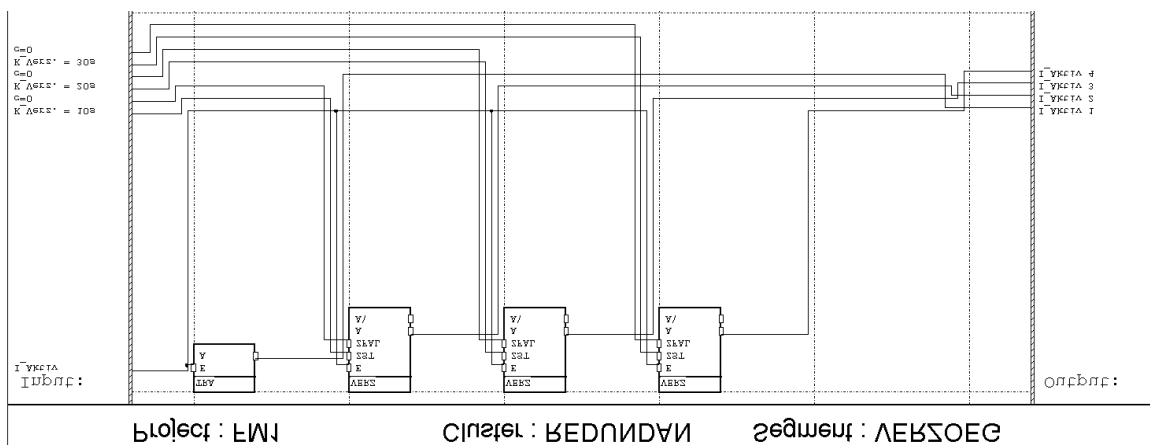
Das Funktionsmodul hat einen Puffer für ausgehende Telegramme, der maximal 100 Telegramme auf einmal zwischenspeichern kann (OS-Version 2.6 und älter). Werden durch die Applikation mehr als 100 Telegramme zum gleichen Zeitpunkt ausgelöst, überläuft dieser Puffer und es gehen Telegramme verloren. (Das Funktionsmodul generiert in diesem Fall eine Exception ID 1.)

In diesem Redundanz-Konzept sollen ja alle Ausgangssignale der ersten Gruppe gesendet werden, wenn das Funktionsmodul aktiv wird. Genau dies führt aber zum Überlaufen des Ausgangspuffers, wenn ein Projekt relativ viele Ausgangssignale hat (über 100). Es bleibt also nichts anderes übrig, als die Telegramme zeitversetzt zu senden.

Dieses Problem kann gelöst werden, indem das Signal 'I\_Aktiv', welches die Telegramme auslöst, verzögert wird und das verzögerte Signal für das Auslösen der Telegramme verwendet wird (siehe untenstehende Schaltung).

Bis der Ausgangspuffer mit 100 Telegrammen wieder leer ist, dauert es mindestens 6 Sekunden. Ist der Bus sonst schon stark belastet, dauert es noch länger. Die Verzögerung der Telegramme muss also nach diesem Kriterium aufgebaut werden.

**Schaltung:**



Die Signale 'I-Aktiv 1' bis 'I\_Aktiv 4' werden in Zeitabständen von 10 Sekunden (= aktiv). Sie können nun anstatt 'I\_Aktiv' für je 100 Ausgangssignale der ersten Gruppe verwendet werden. (Es werden aber besser nur 80 bis 90 Signale miteinander gesendet, um den Ausgangs-Puffer sicher nicht zu überlasten.)

**Festlegen der Zykluszeit**

Bis jetzt wurde immer nur von der Zykluszeit T gesprochen, ohne einen konkreten Wert zu nennen. Die Zykluszeit der Lebenszeichen soll kurz sein, dass ein Ausfall möglichst schnell erkannt werden kann. Doch gibt es Grenzen, die eine beliebig kleine Zykluszeit nicht zulassen. Bei grossen Applikationen, die viele Telegramme



## EIB System

### Funktionsmodul und FMTool

erzeugen, kann der Ausgangs-Puffer stark gefüllt sein. Dadurch wird auch das Senden des Lebenszeichens verzögert. Es macht also keinen Sinn die Zykluszeit kleiner als 10 Sekunden zu wählen.

Eine vernünftige Wahl für die Zykluszeit der Lebenszeichen ist 20 Sekunden. Nach einem Ausfall des aktiven Funktionsmodules dauert es dann zwischen 10 und 30 Sekunden bis das andere Funktionsmodul die Aufgaben übernimmt.

Es ergeben sich folgende Konstanten:

$$T = 20s$$

$$0,5T = 10s$$

$$1,5T = 30s$$

$$2,5T = 50s$$

$$3T = 60s$$

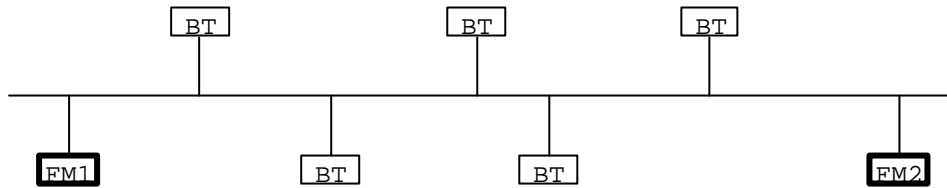
#### Reaktionen beim Ausfall eines Funktionsmodules

- Wenn das aktive Funktionsmodul ausfällt, dauert es zwischen  $0,5T$  und  $1,5T$  bis das passive Funktionsmodul aktiv wird und übernimmt. In dieser Zeit werden keine Funktionen vom Funktionsmodul ausgeführt, aber es empfängt alle Signale. Beim Übernehmen wird es zuerst alle Signale von der ersten Gruppe (Ausgänge bei denen nur eine Änderung des Signalwertes eine Funktion auslöst) auf den Bus senden. Dadurch werden diese Funktionen nachträglich ausgeführt. Schaltet also jemand das Licht im kritischen Moment ein oder ein Lichtsensor schaltet auf dunkel, werden die Reaktionen erst nach dem Übernehmen stattfinden.  
Signale von der zweiten Gruppe (Ausgänge bei denen auch das Senden des gleichen Signalwertes eine Funktion auslöst) werden nicht nachträglich gesandt. Diese Funktionen werden also nicht ausgeführt. Hat also ein Anwender im kritischen Moment versucht eine Jalousie hochzufahren, muss er später erneut drücken.
- Fällt das passive Funktionsmodul aus, sind alle Funktionen jederzeit gewährleistet. Das aktive Funktionsmodul wird den Ausfall an die Visualisierung melden, damit der Schaden behoben werden kann.
- Ein Funktionsmodul, das nur eine kurze Zeit in Betrieb ist, kann die Funktion in den meisten Fällen nicht reibungslos übernehmen, da noch nicht alle Eingangssignale den aktuellen Wert haben.  
Soll z.B. eine neue Applikation in beide Funktionsmodule geladen werden, ist es sinnvoll, zuerst das passive Funktionsmodul zu laden und erst nach einer längeren Zeit das Aktive. Beim Laden des zweiten Funktionsmodules wird so das Erste übernehmen, und der Bus wird fast ohne Unterbruch weiter funktionieren.

Platzierung der Funktionsmodule im Netz:

In einer Linie:

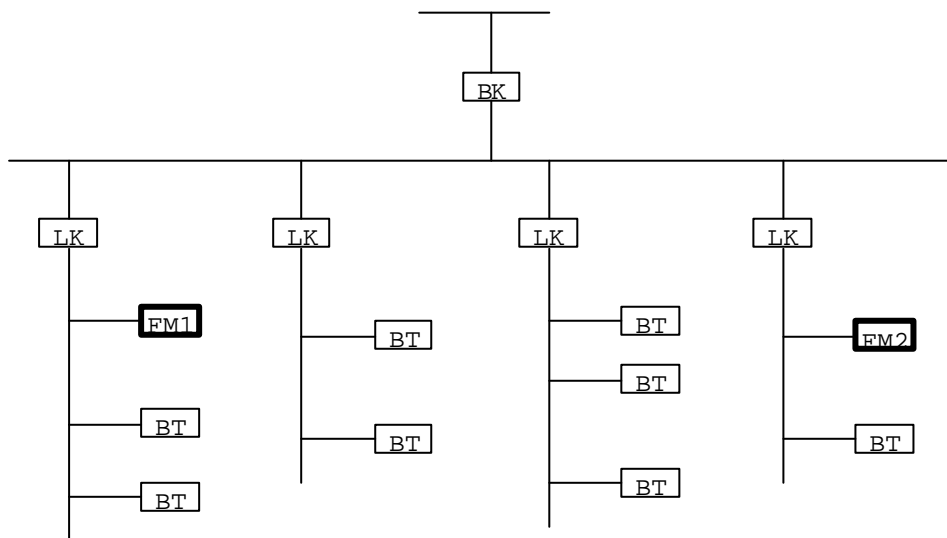
Ist nur eine Linie vorhanden, sollen die Funktionsmodule möglichst weit voneinander entfernt sein. So kann bei einem Leitungsbruch ein Teil des Busses weiter betrieben werden.



BT = Bus-Teilnehmer

In verschiedenen Linien:

Sind verschiedene Linien vorhanden, müssen die Funktionsmodule in verschiedenen Linien, möglichst weit voneinander entfernt, angeschlossen werden. Der Vorteil davon ist, dass sie von separaten Speisungen gespeist werden. Am einfachsten wird bei den Linienkopplern die Filterung ausgeschaltet, weil meistens sowieso fast alle Telegramme über die Linienkoppler zu den Funktionsmodulen durchgeschaltet werden müssen.



BK = Bereichs-Koppler

LK = Linien-Koppler

BT = Bus-Teilnehmer

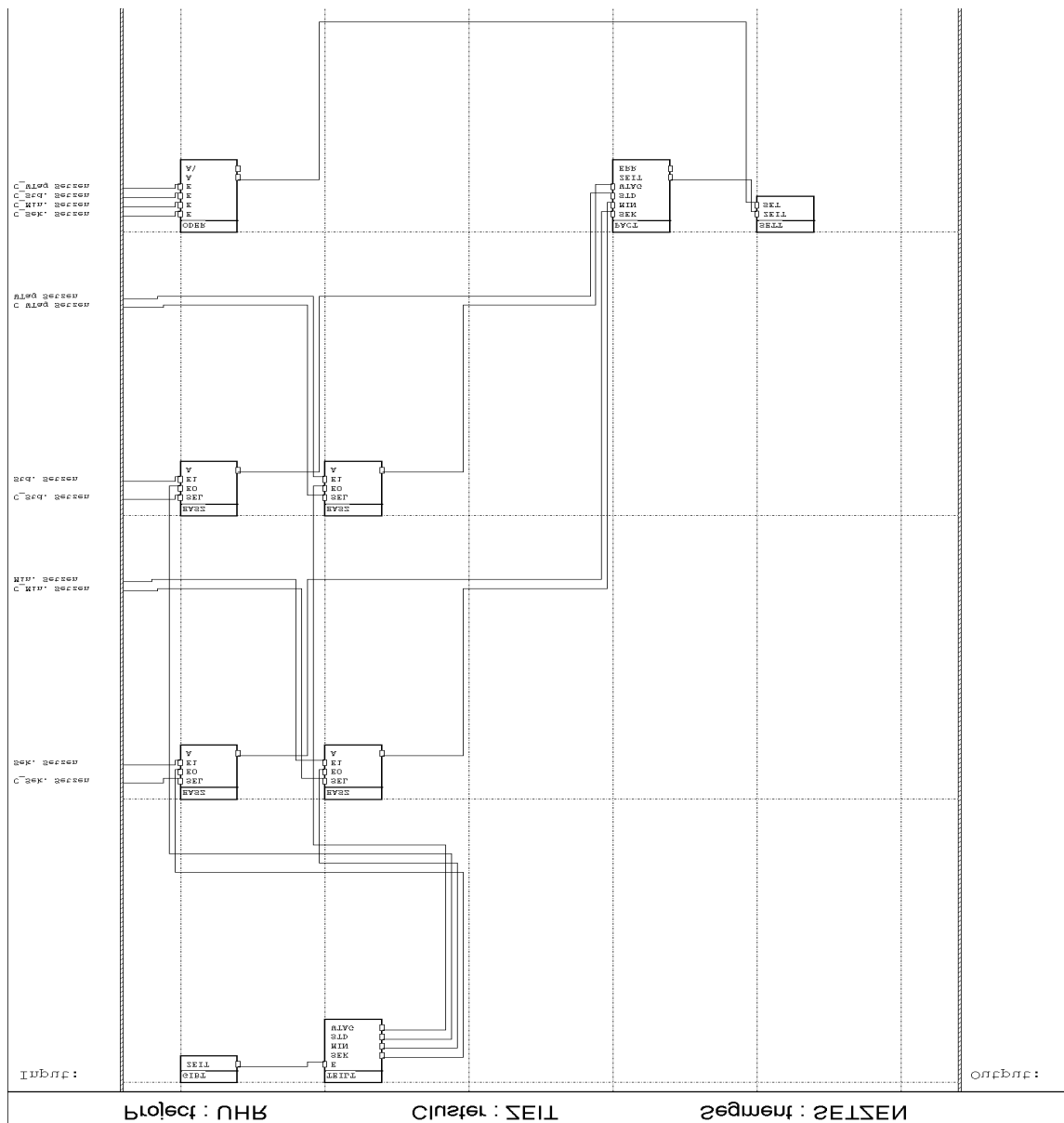
## **14 Zeit setzen**

---

In diesem Beispiel wird gezeigt, wie die interne Uhr des Funktionsmodules bequem, ohne Zahlenformat-Umrechnungen, gesetzt werden kann. Mit der ETS-Inbetriebnahme oder dem Busmonitor können nur Dezimal- oder Hexadezimalzahlen gesendet werden. Um ein Zeit-Telegramm (EIS3), das drei Byte lang ist, zu senden, müssten zuerst Sekunden, Minuten, Stunden und der Wochentag in eine Zahl umgerechnet werden.

In diesem Beispiel können aber die Sekunden, Minuten, Stunden und der Wochentag einzeln gesendet und im Funktionsmodul eingestellt werden. Für jeden dieser Werte existiert eine Gruppenadresse, über welche mit der ETS-Inbetriebnahme eine Dezimalzahl gesendet werden kann.

Schaltung:



In der ersten Zeile wird die aktuelle Zeit in ihre vier Bestandteile aufgeteilt. Diese werden benötigt, damit die Werte, die nicht verstellt werden, unverändert wieder gesetzt werden können.  
 Die vier EASZ-Blöcke übernehmen die gesendeten Werte vom Bus und geben sie auf den PACT-Block, der die vier Werte in ein Zeitsignal umwandelt und auf den SETT-Block gibt.  
 Beispiel: Es wird ein Telegramm auf das Signal 'Min. Setzen' empfangen. Der zweite EASZ-Block schaltet den empfangenen Wert kurz auf seinen Ausgang, weil 'C\_Min. Setzen' kurz 1 ist. Die anderen EASZ-Blöcke schalten die Werte der aktuellen Zeit auf die Ausgänge. Damit liegt nun am ZEIT-Eingang des SETT-Blockes die neue,

## **EIB System**

### **Funktionsmodul und FMTool**

veränderte Zeit an. Der ODER-Block setzt beim Empfangen eines der vier Werte den SET-Eingang des SETT-Blockes auf 1, damit die neue Zeit gesetzt wird.

Zu beachten:

- Der EASZ-Block muss den Datentyp Byte haben.
- Wird eine ungültige Zahl auf einen Eingang gesendet, wird dieser Wert auf 0 gesetzt. Gültige Werte sind:  
Sekunden: 0 .. 59  
Minuten: 0 .. 59  
Stunden: 0 .. 23  
Wochentag: 0..7. 0 = kein Tag (darf nicht gesetzt werden), 1 = Montag, 7 = Sonntag

## 15 Datum setzen

---

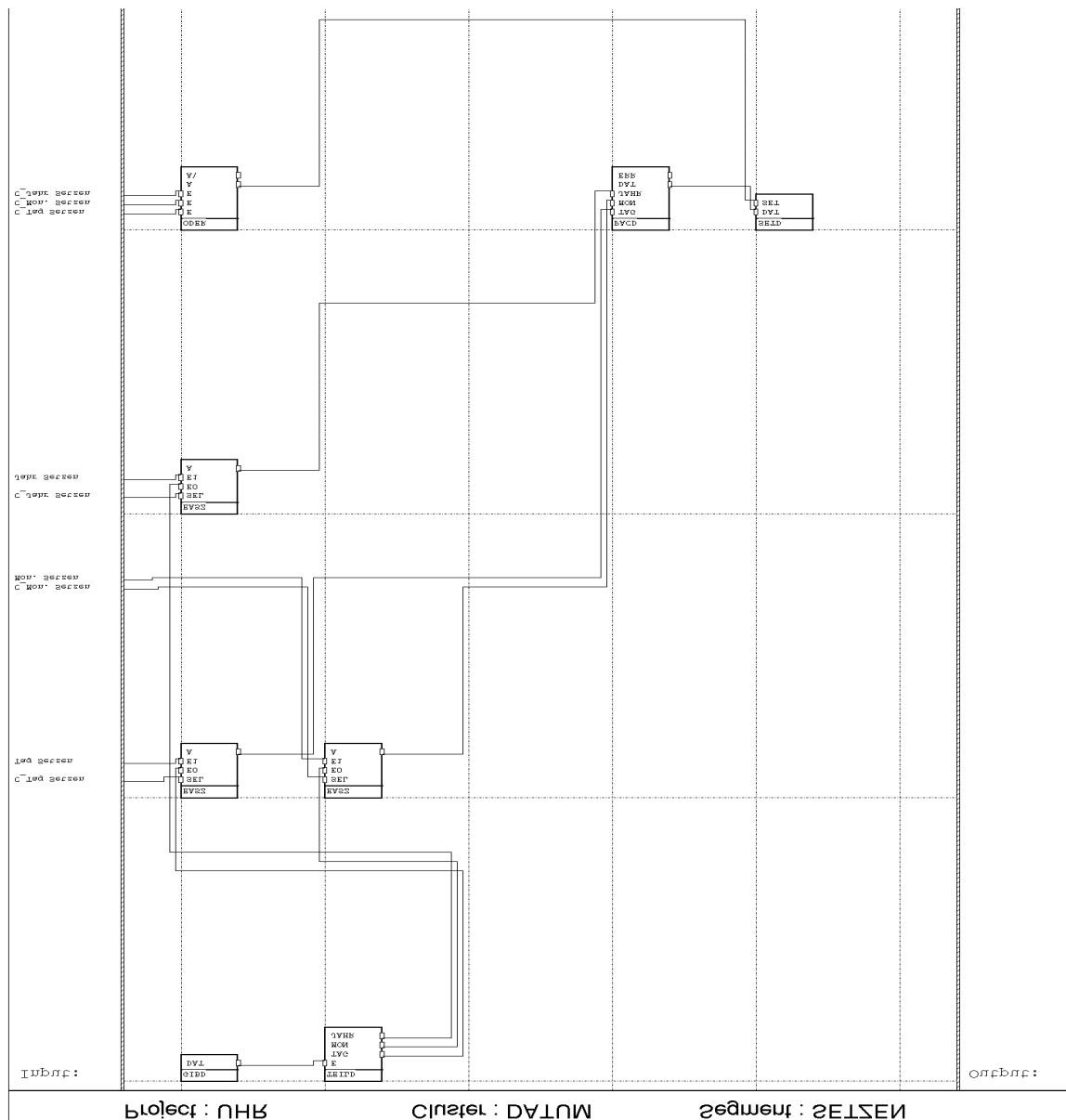
In diesem Beispiel wird gezeigt, wie das Datum der internen Uhr im Funktionsmodul bequem, ohne Zahlenformat-Umrechnungen, gesetzt werden kann. Mit der ETS-Inbetriebnahme oder dem Busmonitor können nur Dezimal- oder Hexadezimalzahlen gesendet werden. Um ein Datum-Telegramm (EIS3), das drei Byte lang ist, zu senden, müssten zuerst Tag, Monat und Jahr in eine Zahl umgerechnet werden.

In diesem Beispiel können aber Tag, Monat und Jahr einzeln gesendet und im Funktionsmodul eingestellt werden. Für jeden dieser Werte existiert eine Gruppenadresse, über welche mit der ETS-Inbetriebnahme eine Dezimalzahl gesendet werden kann.

Schaltung:

# EIB System

## Funktionsmodul und FMTool



In der ersten Zeile wird das aktuelle Datum in seine drei Bestandteile aufgeteilt. Diese werden benötigt, damit die Werte, die nicht verstellt werden, unverändert wieder gesetzt werden können.

Die drei EASZ-Blöcke übernehmen die gesendeten Werte vom Bus und geben sie auf den PACD-Block, der die drei Werte in ein Datumssignal umwandelt und auf den SETD-Block gibt.

Beispiel: Es wird ein Telegramm auf das Signal 'Mon. Setzen' empfangen. Der zweite EASZ-Block schaltet den empfangenen Wert kurz auf seinen Ausgang, weil 'C\_Mon. Setzen' kurz 1 ist. Die anderen EASZ-Blöcke schalten die Werte des aktuellen Datums auf die Ausgänge. Damit liegt nun am DAT-Eingang des SETD-Blockes das

neue, veränderte Datum an. Der ODER-Block setzt beim Empfangen eines der drei Werte den SET-Eingang des SETD-Blockes auf 1, damit das neue Datum gesetzt wird.

Zu beachten:

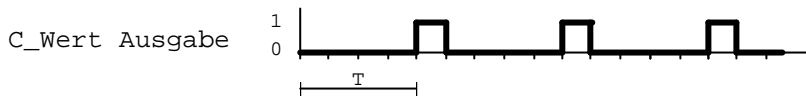
- Der EASZ-Block muss den Datentyp Byte haben.
- Wird eine ungültige Zahl auf einen Eingang gesendet, wird dieser Wert auf 0 gesetzt. Gültige Werte sind:  
Tag: 1 .. 31  
Monat: 1 .. 12  
Jahr: 0 .. 99



## 16 Zyklische Ausgabe von Werten

Soll ein Signalwert zyklisch auf den Bus gesendet werden, kann eine der folgenden Schaltungen angewendet werden.

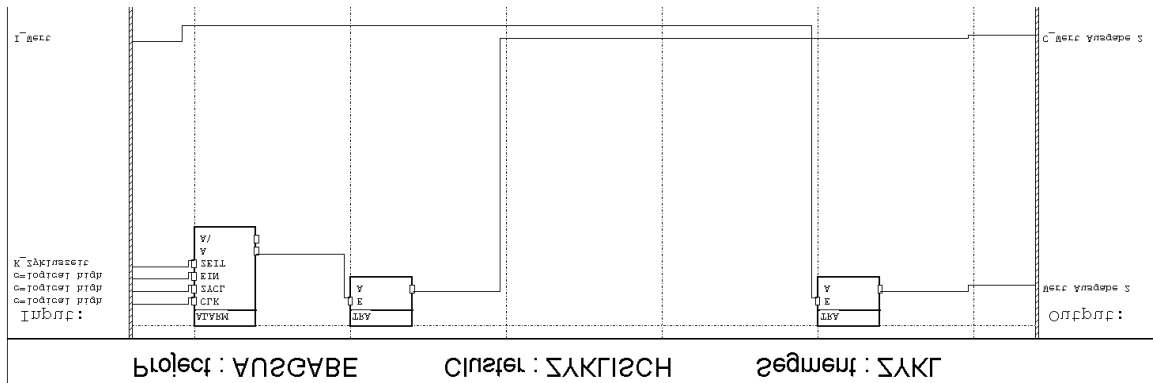
In beiden Schaltungen gibt der ALARM-Block zyklisch einen Puls auf das Kontrollsignal 'C\_Wert Ausgabe'. Dadurch wird der Signalwert zyklisch auf den Bus gesendet.



Weil der CLK-Eingang des ALARM-Blockes auf 1 gesetzt ist, entspricht dies beim Aufstarten des Funktionsmodules einer positiven Flanke und die Alarmfunktion wird sofort gestartet. Der ZYKL- und der EIN-Eingang müssen auf 1 gesetzt sein. Der TRA-Block am Ausgang des ALARM-Blockes ist nötig, da in der FBL-Version 1.04 oder älter, der ALARM-Block nicht direkt verwendet werden kann, um zyklisch auf ein Kontrollsignal zu schreiben.

### Schaltung 1:

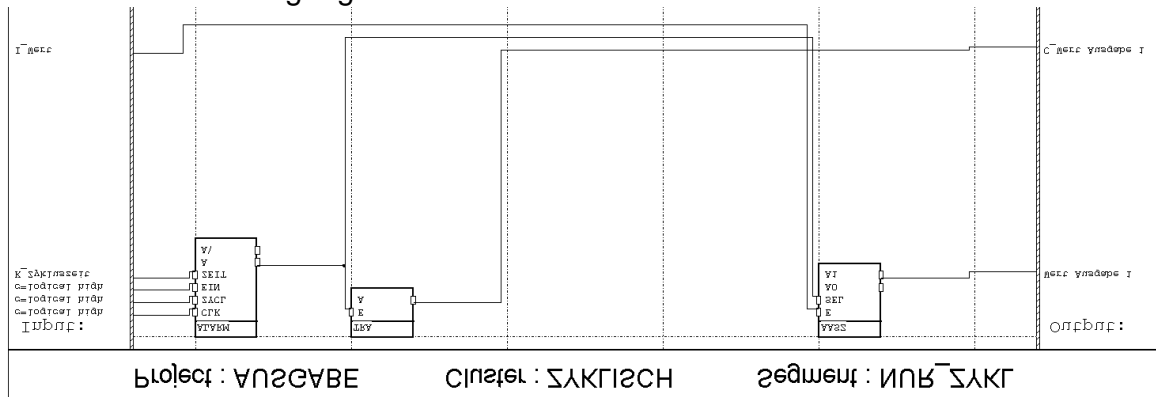
In dieser Schaltung wird der Wert vom Signal 'I\_Wert' zyklisch auf den Bus gesendet. Aber auch wenn sich der Wert ändert, wird ein Telegramm gesendet.



### Schaltung 2:

In dieser Schaltung wird der Wert vom Signal 'I\_Wert' nur zyklisch auf den Bus gesendet. Bei einer Änderung des Wertes wird kein Telegramm gesendet. Der AASZ-Block sorgt dafür, dass der Wert von 'Wert Ausgabe 2' nur ändert, wenn der ALARM-

Block eine 1 am Ausgang hat.



## 17 Zähler für Datentyp Word

In der Funktionsblock-Bibliothek (FBL) Versionen 1.04 oder älter, kann der ZÄHLER-Block nicht mit dem Datentyp Word zählen. In diesem Beispiel wird gezeigt, wie ein Zähler programmiert werden kann, der die gleichen Funktionen hat wie der ZÄHLER-Block, aber am Ausgang den Datentyp Word hat.

Diese Schaltung zählt in der Aufwärtsrichtung von 0 bis 65535 und beginnt dann wieder bei 0. In der Abwärtsrichtung zählt sie im gleichen Zahlenbereich rückwärts. Wie der ZÄHLER-Block, zählt sie nur, wenn am CLK-Eingang eine positive Flanke vorkommt. Sie hat auch einen Reset-Eingang, mit welchem der Ausgang auf 0 gesetzt werden kann.

Wahrheitstabelle:

I_CLK-Eingang	I_Aufwärts (Richtung)	I_Reset	I_Zählerwert (Ausgang)
x	x	1	0
↑	1 (aufwärts)	0	I_Zählerwert + 1
↑	0 (rückwärts)	0	I_Zählerwert - 1

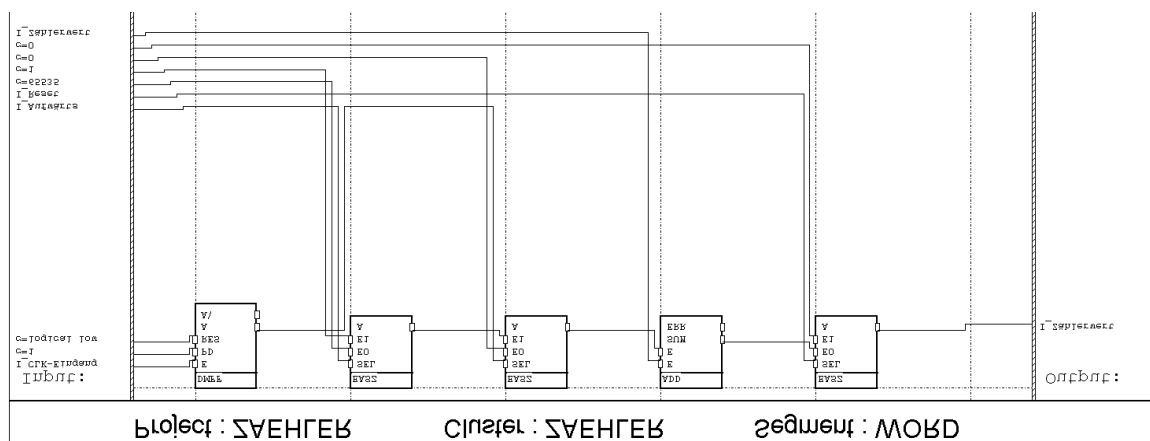
In allen anderen Fällen bleibt der Zählerwert unverändert.

Legende:

x = es hat keinen Einfluss, ob dieser Eingang 0 oder 1 ist

↑ = positive Flanke

Schaltung:



Der DMFF-Block erkennt die positiven Flanken vom Signal 'I\_CLK-Eingang' und gibt dann einen kurzen Puls auf seinen Ausgang. Der erste EASZ-Block schaltet die Zählrichtung um. In Aufwärtsrichtung hat er den Wert 1 am Ausgang, in Abwärtsrichtung 65535. Der zweite EASZ-Block schaltet das Ausgangssignal des ersten EASZ-Blockes auf den einen Eingang des ADD-Blockes, wenn eine positive

Flanke am CLK-Eingang erkannt wurde. Sonst gibt er den Wert 0 weiter. Der ADD-Block addiert nun diesen Wert zum Zählerwert. Der dritte EASZ-Block schaltet, wenn das Reset-Signal 0 ist, die Summe wieder auf den Zählerwert. Der ADD-Block zählt somit 0, 1 oder 65535 zum Zählerwert. Mit 0 verändert sich der Zählerwert nicht, mit 1 wird er um 1 erhöht und mit 65535 wird er um 1 verkleinert. Dass der Zähler beim Addieren von 65535 rückwärts zählt, mag auf den ersten Blick erstauen. Doch dies funktioniert, weil der ADD-Block nur im Zahlenbereich von 0 bis 65535 rechnen kann und bei einem Überlauf wieder bei 0 beginnt (16 Bit-Addition). Wird also zu der Zahl 3 die Zahl 65535 addiert, würde dies 65538 ergeben. Da diese Zahl in 16 Bit nicht dargestellt werden kann, bleibt noch der Rest, nämlich 2, übrig. Der Zählerstand wird also um 1 verkleinert.

Zu beachten:

- Der ADD-Block und die EASZ-Blöcke müssen den Datentyp Word haben.
- Wird die Resetfunktion nicht verwendet, kann der dritte EASZ-Block durch ein TRA-Block ersetzt werden. Es ist aber wichtig, dass der SUM-Ausgang nicht direkt auf das Signal 'I\_Zählerwert' geführt wird, da der ADD-Block so nicht richtig arbeitet (FBL-Versionen 1.04 und älter).
- Wird nur die eine Zählrichtung benötigt, kann der erste EASZ-Block weggelassen werden. An den E1-Eingang des zweiten EASZ-Blockes muss dann die Konstante 1 (für Aufwärts) oder 65535 (für Abwärts) verbunden werden.

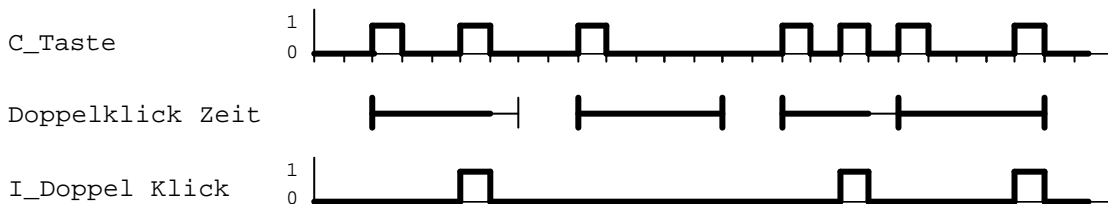
## 18 Erkennen von Doppelklick

In den meisten Fällen kann zum Erkennen von Doppelklicks der KCLICK-Block verwendet werden. Dieser gibt aber das Resultat immer erst nach Ablauf der Erkennungszeit an (Zeit am ZDOP-Eingang).

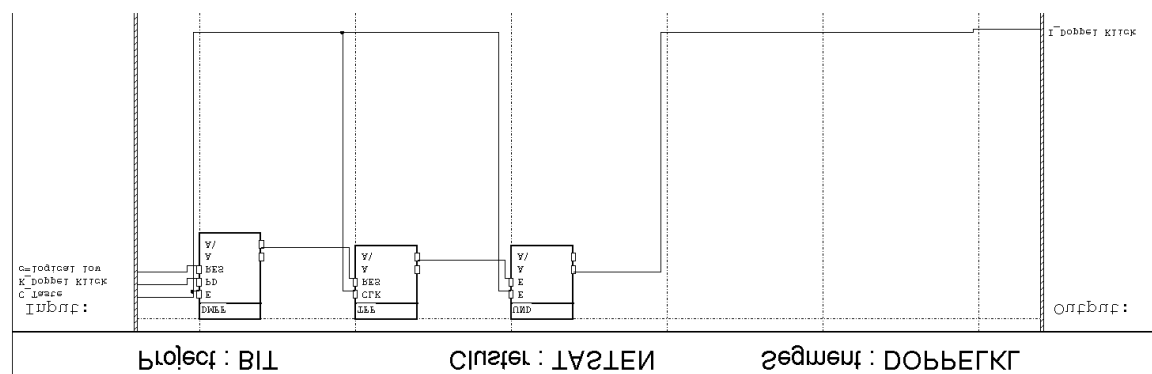
Die hier vorgestellte Schaltung erkennt den Doppelklick sofort wenn der zweite Tastendruck erfolgt. Der zweite Tastendruck muss aber innerhalb einer bestimmten Zeit erfolgen, sonst gilt dies nicht als Doppelklick. Folgen drei Tastendrucke kurz aufeinander, wird der dritte Tastendruck wieder als ersten verstanden, auf welchen ein zweiter Tastendruck folgen kann.

Die Tastendrucke werden anhand des Kontrollsignals der Taste erfasst. Die Taste muss bei jedem Tastendruck nur ein Telegramm auf den Bus senden. Es kommt dabei nicht darauf an, ob es ein Ein- oder ein Ausschalt-Telegramm ist. Es können auch mehrere Tasten auf die gleiche Gruppenadresse senden.

Zeitdiagramm:



Schaltung:



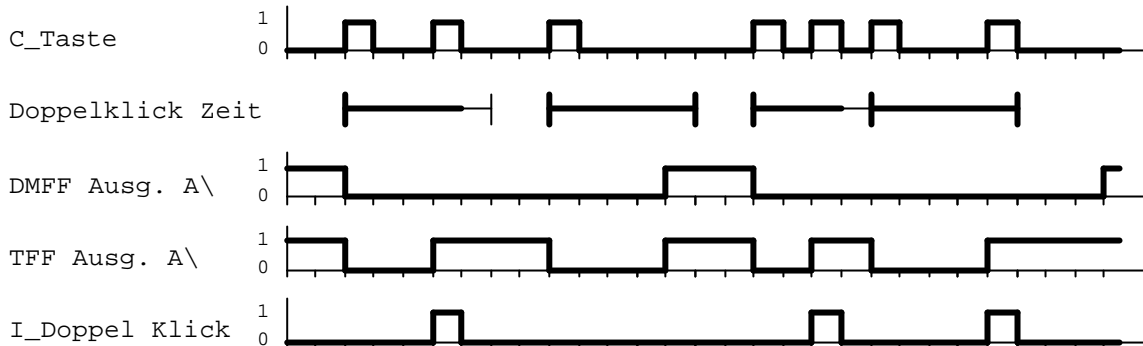
Der DMFF-Block ist für die Zeitmessung zuständig. Am PD-Eingang wird die Zeit eingestellt, in welcher zwei Tastendrucke vorkommen müssen, damit sie als Doppelklick erkannt werden. Bei jedem Tastendruck wird der DMFF-Block gestartet. Kommt nach einem Tastendruck für eine Weile (länger als die eingestellte Zeit am PD-Eingang des DMFF-Blockes) kein neues Telegramm, macht der DMFF-Block ein Reset auf den TFF-Block (A-Ausgang des DMFF-Blockes). Somit ist der A-Ausgang

des TFF-Blockes in einer Pause immer auf 1. Bei jedem Tastendruck wechselt nun dieser Ausgang seinen Zustand. Bei jeder ungeraden Anzahl aufeinanderfolgenden Tastendrucke ist er 0, bei jeder geraden Anzahl ist er 1. Wurde nun eine Taste gedrückt (C\_Taste = 1) und der A\-\-Ausgang des TFF-Blockes ist 1, handelt es sich um einen Doppelklick und das Signal 'I\_Doppel Klick' ist 1. (Siehe detailliertes Zeitdiagramm.)

## EIB System

### Funktionsmodul und FMTool

Detailliertes Zeitdiagramm:



Zu beachten:

- Sind mehrere Tasten auf die Gruppenadresse des Signals 'Taste' konfiguriert, kann es vorkommen, dass ein Doppelklick durch zwei Tasten ausgelöst wird. Auch wenn auf jede Taste nur einmal gedrückt wurde. Ist dies unerwünscht, muss jede Taste eine eigene Gruppenadresse senden und jede einzelne Taste muss mit der hier beschriebenen Schaltung ausgewertet werden.

## 19 Ändern von Einstellwerten zur Laufzeit

Es gibt viele Einstellwerte, die im Prinzip fest bleiben und doch von Zeit zu Zeit verändert werden sollen.

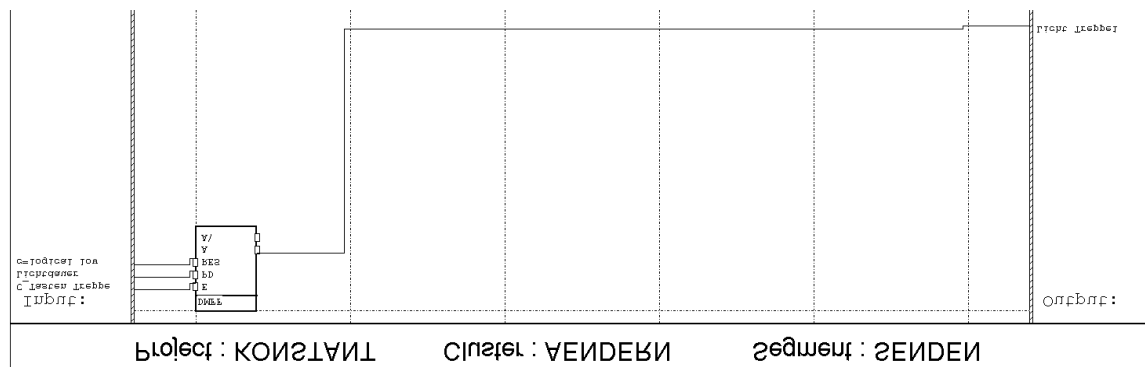
Zum Beispiel:

- Brenndauer des Lichts im Treppenhaus
- Ein- und Ausschaltzeiten von Automatischen Funktionen
- Lichtgrenzwerte, bei denen die Jalousie auf- oder abgefahren wird
- Einstellwerte eines Heizungsreglers

Wird für diese Werte eine Konstante verwendet, können sie nur mit dem FMTool geändert werden. Das Projekt muss dann bei jeder Änderung kompiliert und wieder ins Funktionsmodul geladen werden. Um dies zu umgehen, können die Werte auch über eine Gruppenadresse ins Funktionsmodul geladen werden. Hier sind zwei Vorschläge wie dies realisiert werden kann:

### 1. Senden der Werte

Beispiel:



Dies ist die Schaltung einer einfachen Lichtsteuerung für ein Treppenhaus (siehe auch das Beispiel 'Treppenhaus, einfach'). Die Brenndauer für das Licht ist abhängig vom Wert am PD-Eingang des DMFF-Blockes. Dieser Eingang ist mit dem Eingangssignal 'Lichtdauer' verbunden. Die Gruppenadresse von diesem Signal darf von keinem anderen Gerät am Bus gesendet werden.

Nun kann mit der ETS-Inbetriebnahme ein Wert auf das Signal 'Lichtdauer' gesendet werden und die Lichtdauer auf einen anderen Wert gesetzt werden (siehe 'Telegramm senden' in der ETS-Inbetriebnahme).

Damit nach einem Reset oder einem Spannungsausfall das Signal 'Lichtdauer' einen sinnvollen Wert hat, muss es auf einen vernünftigen Wert initialisiert werden (im Dialog 'Create Signal' oder 'Rename Port Signal' auf 'Set Initial Value' drücken). Ein Signal das nicht initialisiert ist, hat immer den Wert 0.



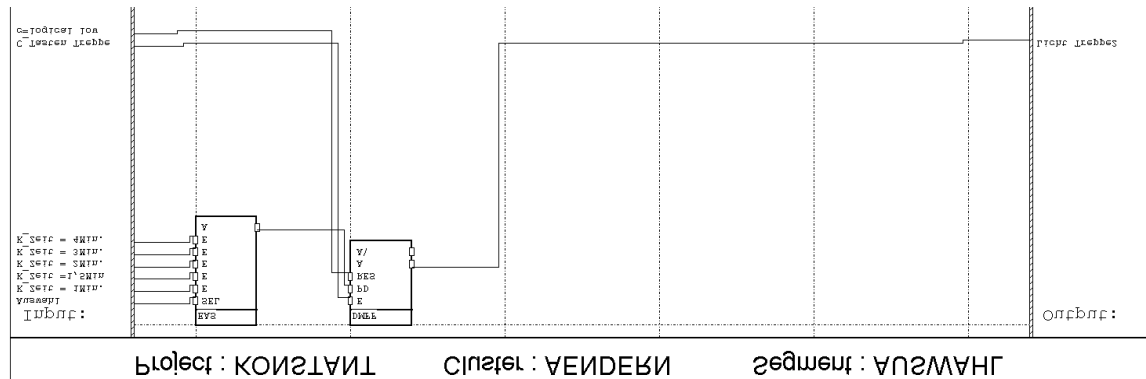
## **EIB System**

### **Funktionsmodul und FMTool**

Diese Methode, einen Einstellwert im Funktionsmodul zu verändern, eignet sich für Werte mit einem Datentyp der ohne Umrechnungen mit der ETS-Inbetriebnahme gesendet werden kann, oder wenn ein Wert genau eingestellt werden muss (Datentypen Word, Sint, Bit). Andernfalls ist die untenstehende Methode zweckmässiger.

## 2. Auswählen von voreingestellten Werten:

Beispiel:



Dieses Beispiel zeigt dieselbe Treppenhaussteuerung wie oben. Doch stammt der Wert für den PD-Eingang nicht von einem Signal, sondern er wird aus einer Reihe von Konstanten ausgewählt und auf den Eingang geschaltet. Die Konstanten sind an den E-Eingängen des EAS-Blockes und werden mit dem Signal 'Auswahl' ausgewählt.

Mit der ETS-Inbetriebnahme wird nur noch die Nummer der gewünschten Konstante auf die Gruppenadresse des Signals 'Auswahl' gesendet. Dies hat den Vorteil, dass nur noch kleine Zahlen gesendet werden müssen und dass Konstanten mit kompliziert zu berechnenden Datentypen (Date, Time, Value) bequem im FMTool eingegeben werden können. Dafür kann der Wert nur in Stufen verändert werden. Diese Schaltung ist also geeignet um Werte zu ändern, die mit der ETS-Inbetriebnahme nur mühsam einzustellen sind.

Zu beachten:

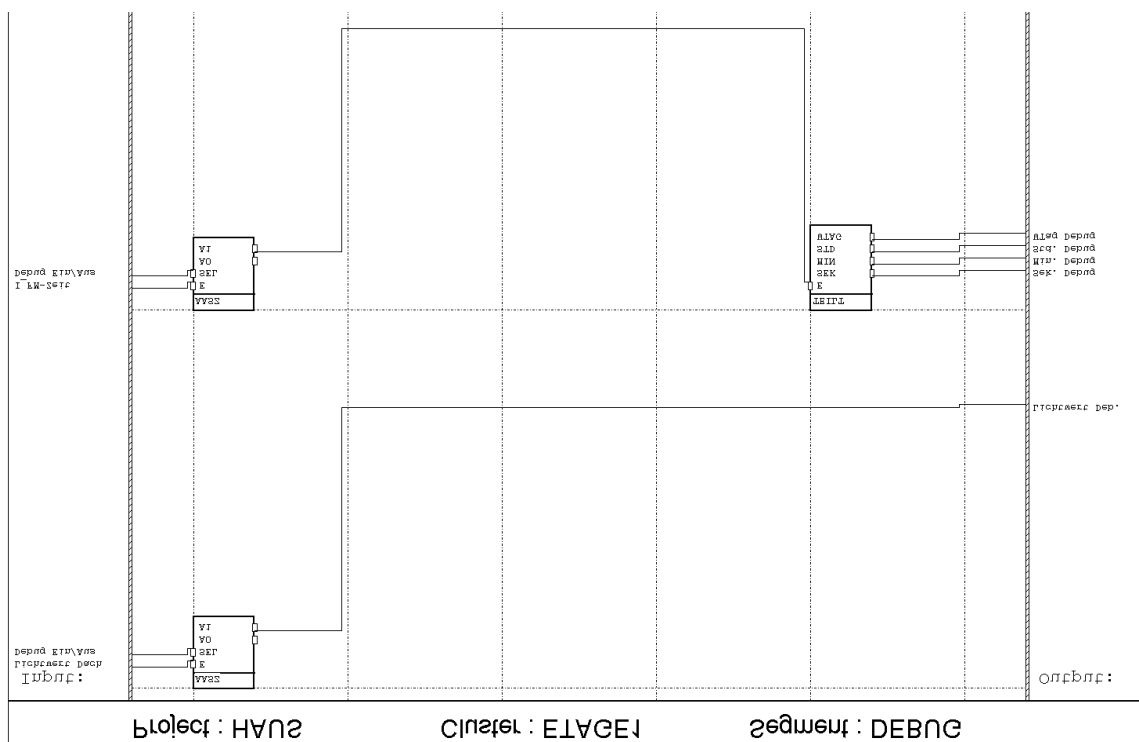
- Die über den Bus eingestellten Werte gehen nach einem Reset verloren. Sie müssen also nach einem Stromausfall im Bus, oder nach dem Download einer neuen Applikation jedesmal neu eingestellt werden.
- Ist eine Visualisierung vorhanden, können die Werte auch über diese eingestellt werden.

## 20 Debuggen

Bei komplizierteren Schaltungen ist es oft schwierig nachzuvollziehen was die Zwischenresultate gerade für Werte haben und ob die Schaltung auch so funktioniert wie man es wünscht. Da es für das Funktionsmodul keinen Debugger gibt, werden hier Möglichkeiten zum Anzeigen von Zwischenresultaten vorgeführt.

Die Zwischenresultate, für die man sich interessiert, müssen jeweils auf ein Ausgangssignal geführt werden. So werden die Zwischenresultate auf den Bus gesendet und können mit der ETS-Inbetriebnahme beobachtet werden.

Schaltung:



In der ersten Zeile wird das interne Signal 'Lichtwert Dach' auf das Ausgangssignal 'Lichtwert Deb.' geschaltet. Die Gruppenadresse von 'Lichtwert Deb.' darf die anderen Busteilnehmer nicht beeinflussen. Es muss also eine Gruppenadresse verwendet werden, die noch frei ist.

Das Funktionsmodul wird nun jede Änderung des Signals auf den Bus senden. Da kein anderer Busteilnehmer diese Telegramme empfängt, werden sie immer dreimal wiederholt. Dies belastet natürlich den Bus. Darum wird das Signal durch den AASZ-Block nur auf den Ausgang geschaltet, wenn das Signal 'Debug Ein/Aus' eingeschaltet ist. 'Debug Ein/Aus' ist ein Eingangssignal und kann mit der ETS-Inbetriebnahme oder durch einen Schaltsensor ein- und ausgeschaltet werden. Wenn

nicht überwacht wird, kann so die Debugfunktion ausgeschaltet werden, und der Bus wird nicht unnötig belastet.

In der zweiten Zeile wird gezeigt, wie Signale mit einem komplizierteren Datentyp auf den Bus gesendet werden können. Die Zeit wird, nachdem sie mit dem AASZ-Block ein- und ausgeschaltet wurde, mit dem TEILT-Block in Sekunden, Minuten, Stunden und Wochentag aufgeteilt. Dadurch müssen die empfangen Telegramme bei der Auswertung nicht umgerechnet werden.

## EIB System

### Funktionsmodul und FMTool

#### Überwachung der Signale

Um die Änderungen der Signale zu überwachen gibt es mehrere Möglichkeiten. Hier die wichtigsten drei:

#### ETS-Inbetriebnahme

Mit der ETS-Inbetriebnahme können die Telegramme auf dem Bus aufgezeichnet werden und nachher analysiert werden. So können die Reaktionen des Funktionsmodules auf bestimmte Bus-Ereignisse genau nachvollzogen werden. Auch zeitabhängige Funktionen können gut getestet werden, da bei jedem empfangenen Telegramm die Zeit festgehalten ist.

Ändern die Werte nur langsam, kann mit der ETS-Inbetriebnahme auch ein einzelnes Ausgangssignal abgefragt werden. (Siehe Menü 'Diagnose' in der ETS-Inbetriebnahme)

#### EITT (EIB Interoperability Test Tool)

Das EITT ist ein Tool für Entwickler, doch es hat Funktionen, die auch für den Anwender sehr nützlich und einfach zu bedienen sind. Während den Aufzeichnungen werden die empfangenen Telegramme laufend am Bildschirm angezeigt. Es kann auch jederzeit ein Telegramm generiert werden. Es können auch ganze Telegrammsequenzen programmiert und abgespielt werden. Die Telegramme können ohne Umrechnungen in jedem EIS-Format eingegeben werden.

#### Schaltensor

Wenn nur Bit-Signale angezeigt werden sollen, kann eine einfache 'Visualisierung' mit einem Schaltsensor realisiert werden. Der Schaltsensor muss mit der entsprechenden Gruppenadresse programmiert werden und die LED's müssen den Zustand des Schaltobjekts anzeigen. So kann der Zustand des Signales auf eine einfache und bequeme Art überwacht werden.

#### Tips:

- Um schnelle Abläufe besser verfolgen zu können, kann die Taskzykluszeit im Taskbuilder vorübergehend grösser gewählt werden.
- Um langsame Abläufe effizienter debuggen zu können, kann die Taskzykluszeit im Taskbuilder vorübergehend kleiner gewählt werden.
- Werden komplizierte Aufgaben mit dem Funktionsmodul gelöst, empfiehlt es sich, zuerst nur Teilprobleme davon zu programmieren und in einem kleinen Laboraufbau auszutesten. So lassen sich Fehler einfacher finden.
- Wird die oben gezeigte Debug-Schaltung in ein separates Segment gezeichnet, kann dieses Segment nach dem Testen wieder aus dem Task entfernt werden.

## 21 Zeitabhängige Steuerungen (SUHR-Block)

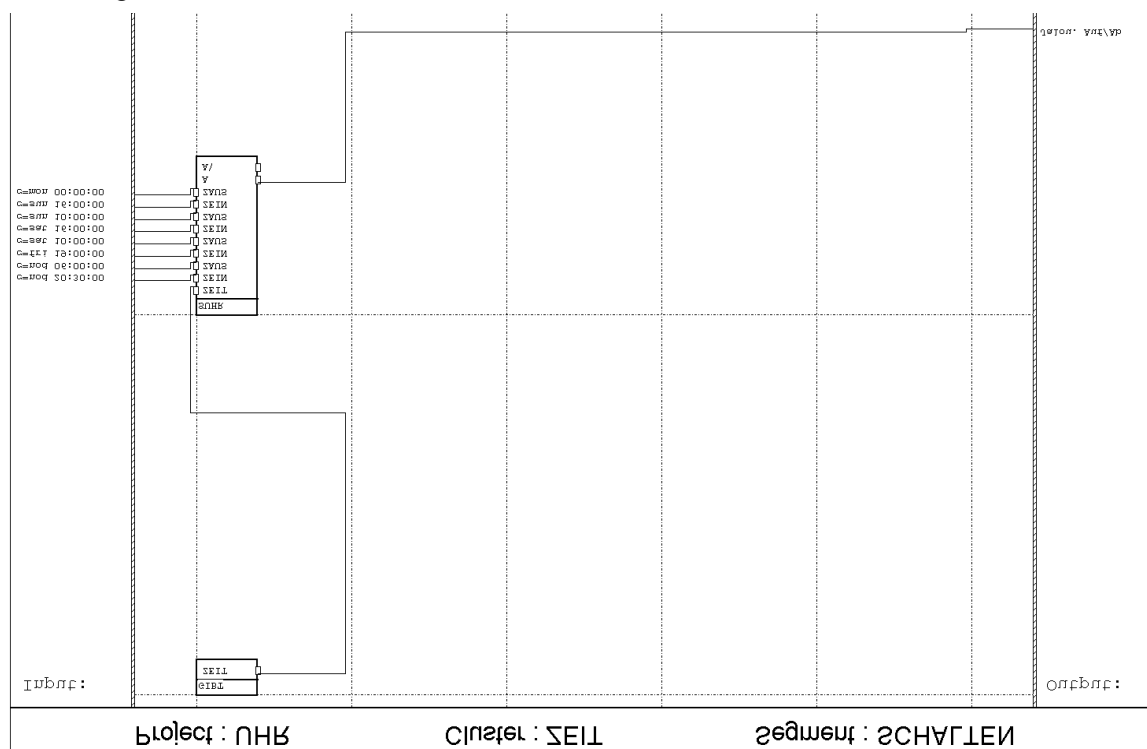
Dieses Beispiel zeigt, wie der Funktionsblock SUHR angewendet wird.

Beispiel:

Die Jalousien in einem Geschäftsgebäude soll automatisch in der Nacht heruntergelassen werden und am Morgen wieder aufgemacht werden. An den Arbeitstagen soll sie jeweils von 6.00 Uhr bis 20.30 Uhr oben sein. Am Freitag soll sie aber schon um 19.00 Uhr zugemacht werden. Am Samstag und am Sonntag soll sie zwischen 10.00 Uhr und 16.00 Uhr oben sein, damit die Zimmerpflanzen auch am Wochenende etwas Licht erhalten.

Diese Aufgabe kann ganz einfach mit dem SUHR-Block realisiert werden:

Schaltung:



Die Jalousie wird über das Signal 'Jalou. Auf/Ab' gesteuert. Wird dieses Signal 1, geht die Jalousie herunter, wird es 0, geht die Jalousie auf. Ist die Zeit in einem der Einschaltbereiche des SUHR-Blockes, geht die Jalousie also zu.

Der erste Einschaltbereich (nod 20:30:00 bis nod 06:00:00) sorgt dafür, dass die Jalousie jede Nacht von 20.30 Uhr bis 6.00 Uhr unten ist. Die Einschaltbereiche (fri 19:00:00 bis sat 10:00:00) und (sat 16:00:00 bis sun 10:00:00) "verlängern" die Nacht

## **EIB System**

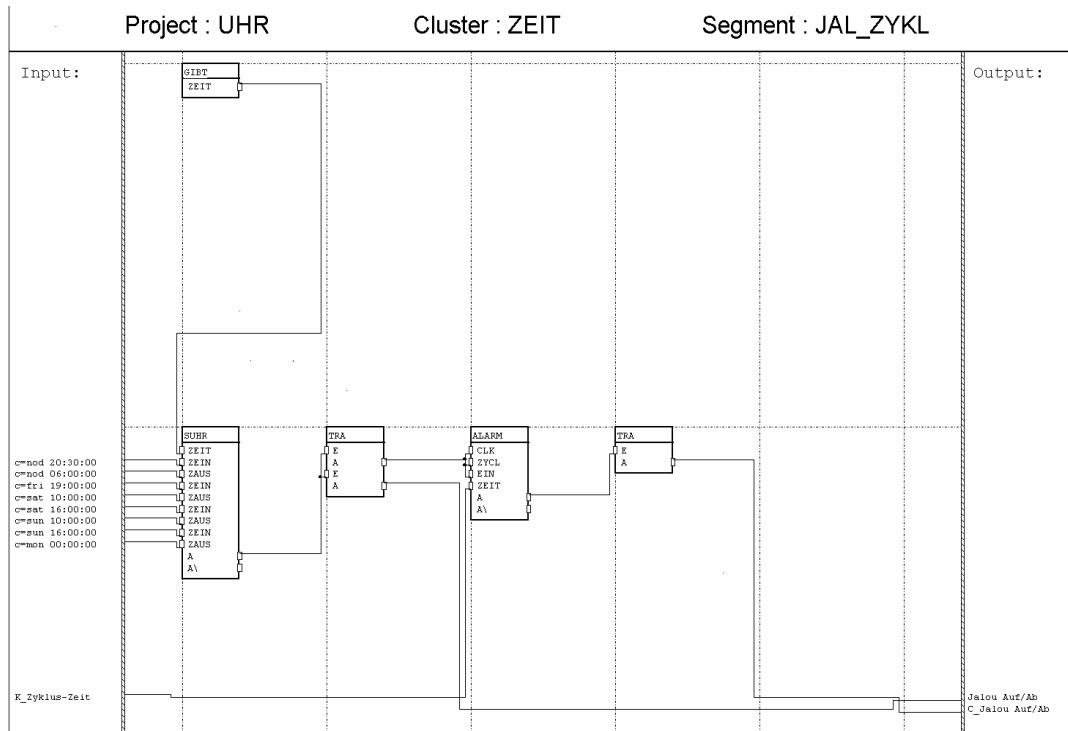
### **Funktionsmodul und FMTool**

auf den Samstag und auf den Sonntag. Der letzte Einschaltbereich (sun 16:00:00 bis mon 00:00:00) schliesst die Jalousie am Sonntag schon um 16.00 Uhr.

#### **Zyklisches Schliessen**

Im obigem Beispiel wird das Telegramm zum Schliessen der Jalousien nur einmal am Abend gesendet. Öffnet nun jemand von Hand die Jalousie, bleibt sie bis zum nächsten Tag oben, wenn sie nicht wieder von Hand herunter gelassen wird. Wir können dieses Problem lösen, indem wir das Zu-Telegramm (hat den Wert 1) zyklisch senden.

Schaltung:



Der erste TRA-Block kopiert das Ausgangssignal des SUHR-Blockes in ein lokales Signal für den ALARM-Block und in ein externes Signal für das Steuern der Jalousie. Wird das Signal 1 (Jalousie fährt herunter) gehen der CLK-, ZYKL- und EIN-Eingang des ALARM-Blockes auf 1. Dadurch wird dieser Block gestartet, und er gibt zyklisch einen Puls auf das Ausgangssignal 'C\_Jalous Auf/Ab'. Ist der Ausgang des SUHR-Blockes 0, wird der ALARM-Block gestoppt (der EIN-Eingang ist 0).

Der TRA-Block am Ausgang des ALARM-Blockes ist nötig, da in der FBL-Version 1.04 oder älter, der ALARM-Block nicht direkt verwendet werden kann, um zyklisch auf ein Kontrollsignal zu schreiben.

Die Zykluszeit 'K\_Zyklus-Zeit' wird z.B. auf eine halbe Stunde oder eine Stunde eingestellt. Diese Schaltung schliesst also zyklisch die Jalousie in der Nacht und öffnet sie am Morgen. Am Tag ist die Jalousie frei von Hand bedienbar.

Tips:

- Der SUHR-Block hat bis zu sieben Einschaltbereiche. Damit kann jeder Wochentag unterschiedlich programmiert werden.
- Werden mehr als sieben Einschaltbereiche benötigt, können mehrere SUHR-Blöcke verwendet werden. Die A-Ausgänge müssen dann mit einem ODER-Block verknüpft werden.



**EIB System**  
**Funktionsmodul und FMTool**

## 22 Datumsabhängige Steuerungen

---

Dieses Beispiel zeigt, wie das Systemdatum des Funktionsmodules eingesetzt werden kann, um datumsabhängige Funktionen zu realisieren.

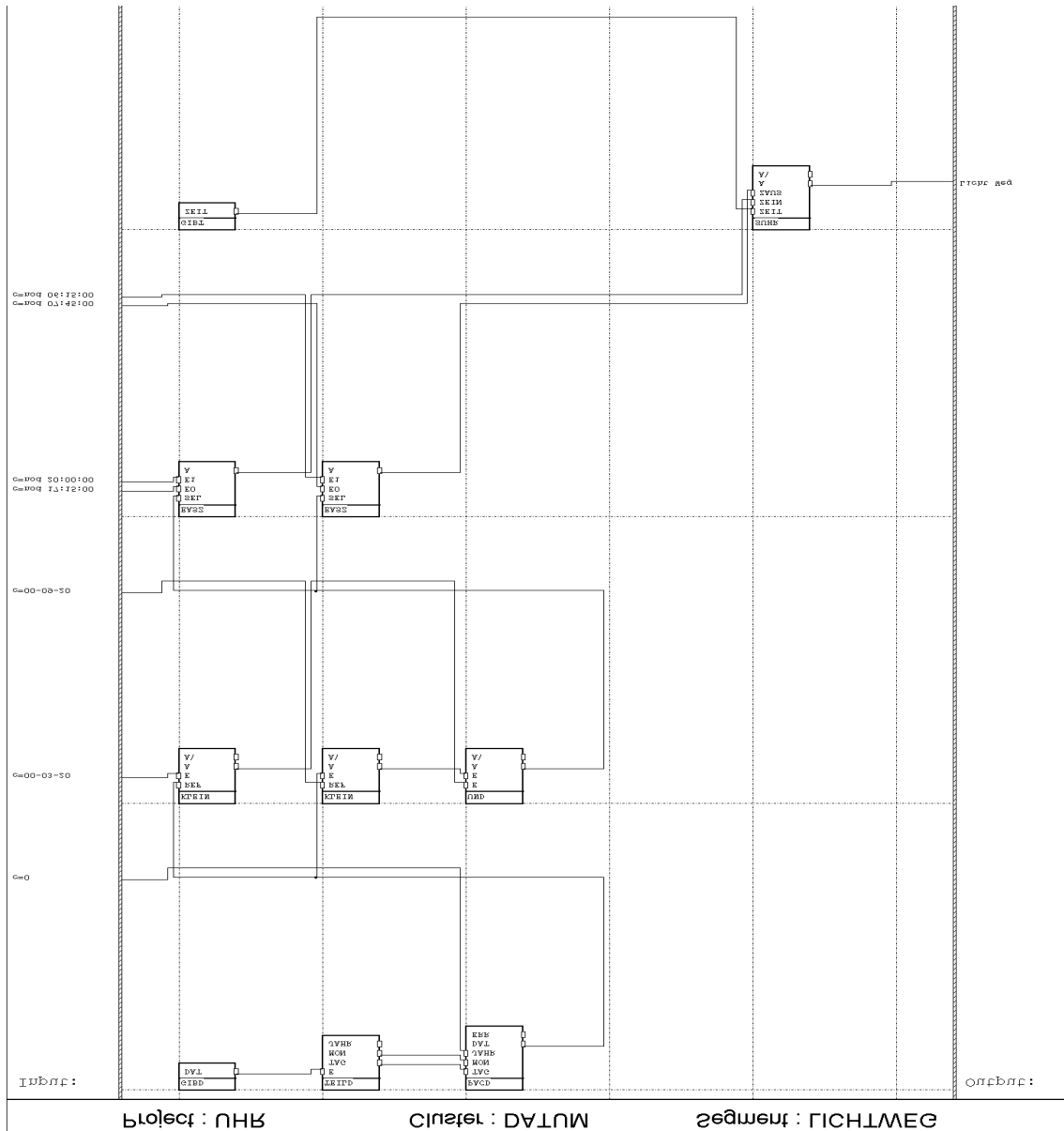
Beispiel:

Die Beleuchtung eines Fussweges zu einem Haus soll immer nur in der Nacht brennen. Wegen der Sommerzeitschaltung und den kurzen Nächten im Sommer, werden unterschiedliche Ein- und Ausschaltzeiten verwendet. Im Winterhalbjahr soll das Licht von 17.15 Uhr bis 7.45 Uhr brennen und im Sommer von 20.00 Uhr bis 6.15 Uhr. Die Umschaltung soll am 20. März und am 20. September stattfinden. Ungefähr an diesen Daten wird auch die Uhrzeit auf die Sommerzeit umgeschaltet.

Schaltung:

# EIB System

## Funktionsmodul und FMTool



Die KLEIN-Blöcke in der zweiten Zeile prüfen, ob wir uns im Sommerhalbjahr befinden oder nicht. Da das Kalenderjahr für diesen Vergleich unwichtig ist, wird in der ersten Zeile das Jahr auf 0 gesetzt. (Der GIBD-Block liefert das aktuelle Datum und mit dem TEILD- und dem PACD-Block wird das Jahr des DAT-Ausganges fix auf 0 gesetzt.)

Der erste KLEIN-Block prüft, ob das Datum grösser als 20. März ist, und der zweite KLEIN-Block stellt fest, ob das Datum kleiner als 20. September ist. Trifft beides zu, ist der Ausgang des UND-Blockes 1 (= Sommerhalbjahr).

Die beiden EASZ-Blöcke schalten entweder die Schaltzeiten für den Sommer oder für den Winter auf den ZEIN- und den ZAUS-Eingang des SUHR-Blockes. Im Winter

schaltet der SUHR-Block mit den Zeiten an den E0-Eingängen, im Sommer mit den Zeiten an den E1-Eingängen.

Der GIBT-Block gibt die aktuelle Zeit auf den SUHR-Block.

Tips:

- Die KLEIN-Blöcke müssen den Datentyp Date haben, die EASZ-Blöcke den Datentyp Time.
- Das Umschalten der Systemuhr auf Sommerzeit, kann einfach mit der Schaltung im Beispiel 'Zeit setzen' gemacht werden.
- Dieses Beispiel teilt das Jahr in nur zwei Abschnitte. Um im Sommer mehr Strom zu sparen, können natürlich auch mehr Abschnitte gemacht werden. Eventuell ist es sinnvoller, die Ein- und Ausschaltzeiten für das Licht anhand eines Lichtsensors zu schalten.

## 23 Vergleichen von Werten mit Hysterese

---

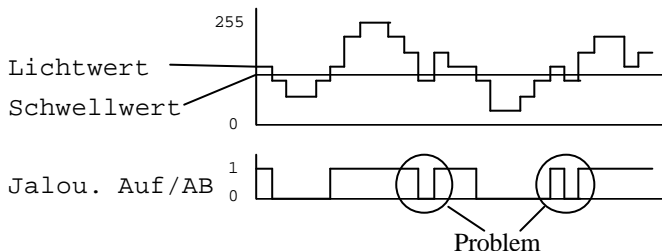
Werden analoge Messwerte wie die Helligkeit oder die Temperatur zum Schalten von Aktoren verwendet, müssen diese mit Schwellwerten verglichen werden.

Als Beispiel wird hier eine Jalousie abhängig vom Aussenlicht gesteuert. Ein Lichtsensor sendet zyklisch einen Lichtwert auf den Bus. Die Jalousie soll automatisch heruntergefahren werden wenn der Lichtwert einen bestimmten Schwellwert übersteigt und soll geöffnet werden, wenn der Schwellwert wieder unterschritten wird.

Wie das erste Zeitdiagramm (ohne Hysterese) zeigt, kann es Probleme geben, wenn nur ein Schwellwert verwendet wird. Ist nämlich der Lichtwert kurzen Schwankungen unterworfen – z.B. wenn kleine Wolken vor der Sonne vorbeiziehen – so wird die Jalousie unnötig auf- und abgefahren.

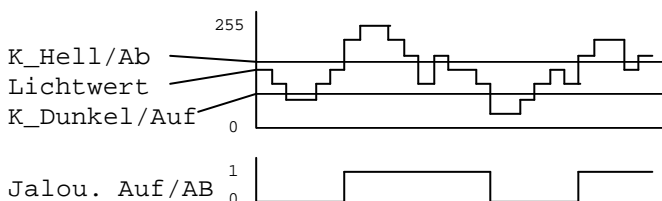
Dieses Problem kann behoben werden, wenn zwei unterschiedliche Schwellwerte definiert werden. Der eine Schwellwert gibt an bei welchem Lichtwert die Jalousie heruntergelassen wird, der andere Schwellwert gibt an wann die Jalousie geöffnet werden soll. Sind die beiden Schwellwerte genug weit auseinander, werden kleinere Schwankungen und Sprünge des Lichtwertes ignoriert (siehe Zeitdiagramm mit Hysterese).

Zeitdiagramm ohne Hysterese (schlecht):



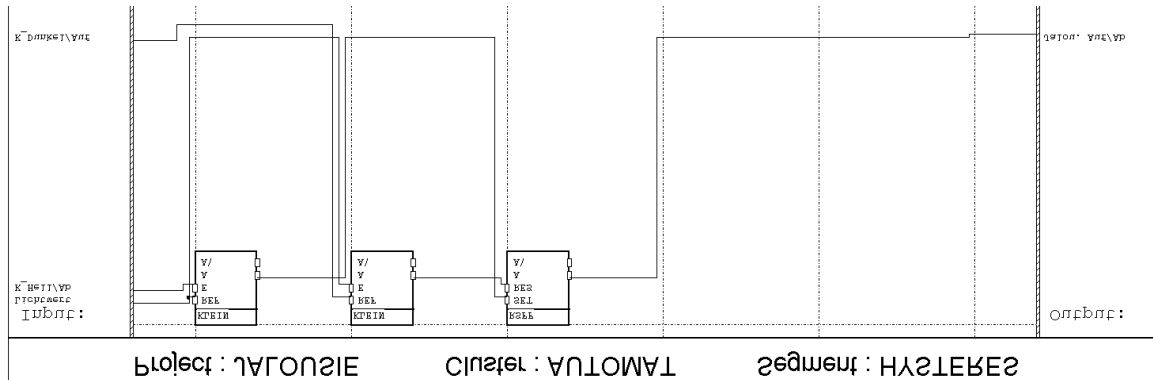
Wird "Jalou. Auf/Ab" 1, so wird die Jalousie heruntergelassen, schaltet das Signal auf 0, wird die Jalousie geöffnet.

Zeitdiagramm mit Hysterese (gut):



Wird "Jalou. Auf/Ab" 1, so wird die Jalousie heruntergelassen, schaltet das Signal auf 0, wird die Jalousie geöffnet.

Schaltung:



Der erste KLEIN-Block vergleicht den Lichtwert mit dem oberen Schwellwert "K\_Hell/Ab". Ist der Lichtwert höher als der Schwellwert, wird der RSFF-Block auf 1 gesetzt und somit die Jalousie heruntergefahren. Der zweite KLEIN-Block vergleicht den Lichtwert mit dem unteren Schwellwert. Sinkt der Lichtwert unter diesen Schwellwert, wird der RSFF-Block auf 0 gesetzt (Reset). Dies öffnet die Jalousie. Bewegt sich nun der Lichtwert zwischen den beiden Schwellwerten, so wird der RSFF-Block weder auf 1 noch auf 0 gesetzt und er bleibt im aktuellen Zustand. Dieser "tote" Bereich wird Hysterese genannt.

Wichtig:

- Damit eine Hysterese entsteht, muss "K\_Hell/Ab" grösser als "K\_Dunkel/Auf" sein.
- Die oben gezeigte Schaltung kann Signale mit verschiedenen Datentypen verarbeiten. Je nach Datenformat des verwendeten Fühlers müssen die KLEIN-Blöcke auf den entsprechenden Datentyp eingestellt werden (Byte, Word, Sint, Value).

## 24 Automatisches Berechnen von Zeiten bei Funktionsblöcken

---

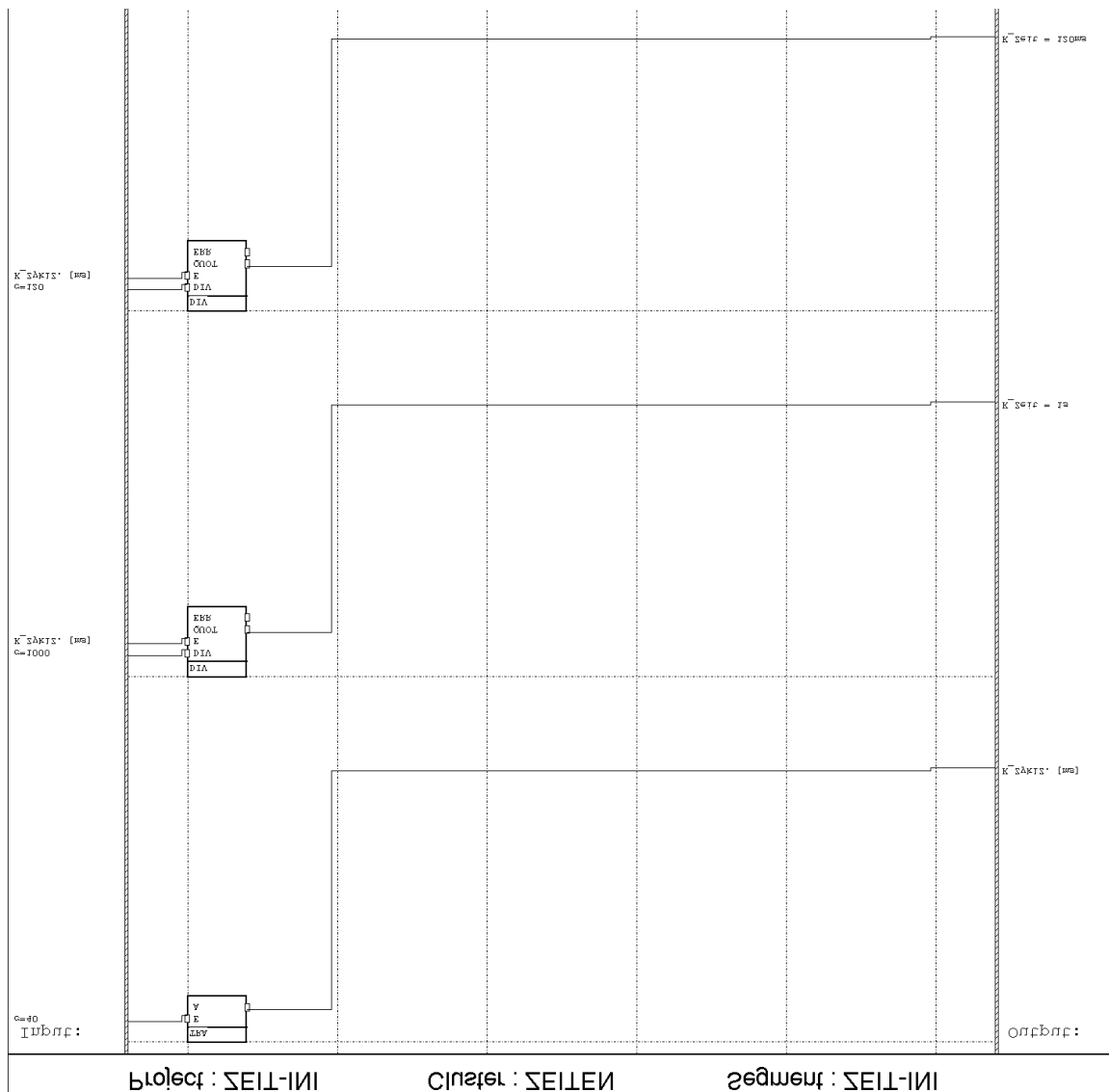
Alle Zeitkonstanten für zeitabhängige Funktionsblöcke (wie z.B. ALARM, DMFF, VERZ) müssen in Anzahl Taskzyklen angegeben werden. Dies kann lästig sein, da der Wert für den Eingang immer berechnet werden muss und die eingestellte Zeit nicht auf den ersten Blick ersichtlich ist. Das folgende Beispiel vereinfacht diese Situation.

Die Formel die zur Berechnung der Zeitkonstanten der zeitabhängigen Funktionsblöcke verwendet wird, soll das Funktionsmodul selber berechnen.

Formel: 
$$\text{Zeitdauer} = \frac{\text{Zeitkonstante [ms]}}{\text{Taskzykluszeit [ms]}}$$

So können die Werte für die Zeitkonstanten bequem in Millisekunden eingegeben werden.

Schaltung:



Der TRA-Block speichert die Konstante, die die Taskzykluszeit angibt, in das interne Signal 'K\_ZyklZ. [ms]'. Die Taskzykluszeit muss also nur an einer Stelle im Projekt angegeben werden und muss nur einmal auf den Wert gesetzt werden, der im Taskbuilder eingestellt ist.

Die DIV-Blöcke (vom Datentyp Word) berechnen nun die Zeitwerte für die zeitabhängigen Funktionsblöcke. Das Resultat wird in interne Signale gespeichert. In diesem Beispiel sind dies die Signale 'K\_Zeit = 1s' und 'K\_Zeit = 120ms'. Die Bezeichnung 'K\_' wurde gewählt, damit sofort ersichtlich ist, dass es sich hier um ein internes Signal handelt, das eigentlich eine Konstante ist und seinen Wert nie ändert. Wird nun die Taskzykluszeit im Taskbuilder verändert, muss nur der Wert von 'K\_ZyklZ. [ms]' geändert werden und alle Zeitkonstanten haben automatisch den richtigen Wert.

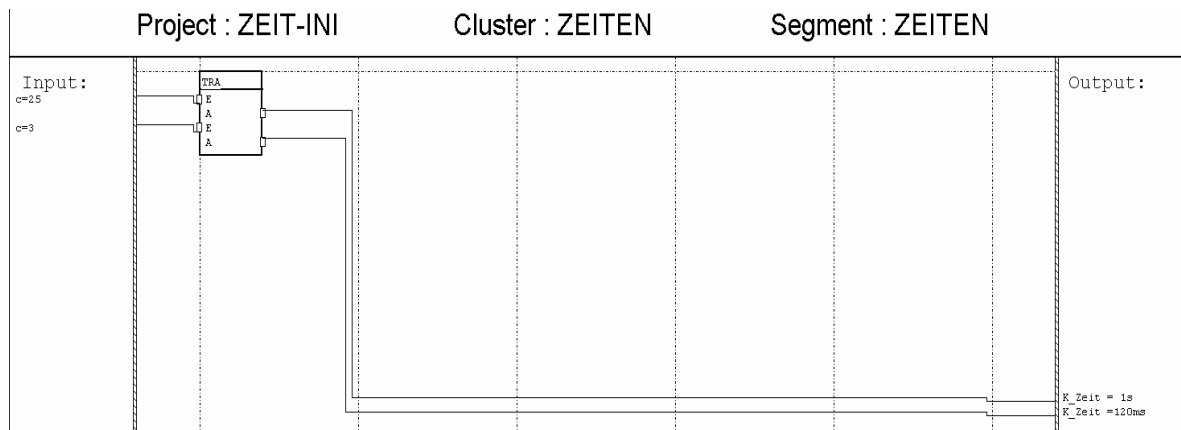


## EIB System

### Funktionsmodul und FMTool

Wichtig:

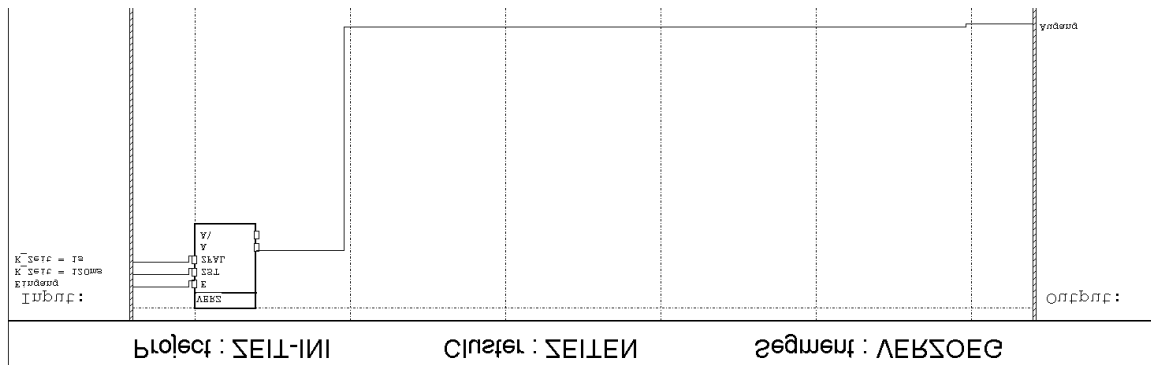
- Das obenstehende Segment soll dem Initialisierungstask zugeordnet werden. Damit werden die Zeitkonstanten nur einmal berechnet und das Funktionsmodul nicht jeden Taskzyklus belastet.
- Es ist klar, dass nur Zeitkonstanten realisiert werden können, die durch die Zykluszeit teilbar sind. Die Konstanten an den DIV-Eingängen müssen also ganzzahlig durch die Taskzykluszeit dividiert werden können.
- Da die Konstanten an den DIV-Eingängen durch die Taskzykluszeit dividiert wird, können lange Zeiten nicht mehr eingestellt werden. Der Datentyp Word kann nur Zahlen bis 65535 darstellen. Es können also nur Zeitkonstanten bis 65520 ms realisiert werden (dies ist nur 65 Sekunden). Dem kann abgeholfen werden, indem die Zeitkonstanten nicht in Millisekunden, sondern in Hundertstel-Sekunden angegeben werden. Die Taskzykluszeit muss dann natürlich auch in Hundertstel-Sekunden definiert werden.
- Eine andere Möglichkeit ist, die Zeitkonstanten nicht mit dem DIV-Block zu berechnen, sondern sie alle in einem Segment zu definieren und in ein internes Signal zu kopieren. So können alle Änderungen in einem Segment vorgenommen werden. Dies hat z.B. Vorteile, wenn die Taskzykluszeit



verändert wird und alle Zeitkonstanten geändert werden müssen.

Anwendungsbeispiel:

Die untenstehende Schaltung zeigt, wie die internen Signale 'K\_Zeit = 1s' und 'K\_Zeit = 120ms' verwendet werden.



Tips:

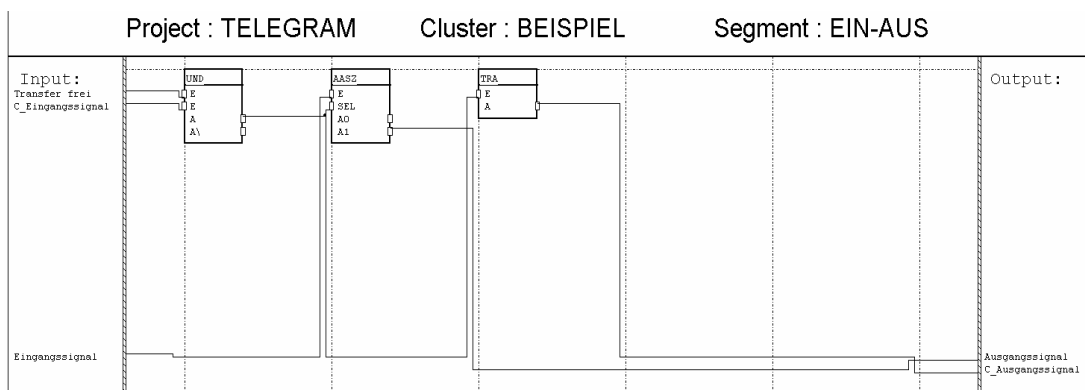
- Es ist empfehlenswert, oft verwendete Konstanten nur an einem Ort im Projekt zu definieren und in ein internes Signal zu speichern. So müssen Änderungen nur einmal gemacht werden. Ausserdem ist es auch übersichtlicher, wenn an einem Blockeingang nicht nur eine Zahl steht, sondern ein Signalname, der die Bedeutung der Konstante angibt. Zum Beispiel: K\_Lichtdauer, K\_Tastenzzeit, K\_Wartezeit.

## 25 Telegrammfluss ein- und ausschalten

Oft soll die Funktion von bestimmten Busgeräten (z.B. Taster) abgeschaltet werden können. Dieses Beispiel zeigt, wie dies mit dem Funktionsmodul gemacht werden kann.

Die Gruppenadresse der abzuschaltenden Funktion muss auf ein Eingangssignal des Funktionsmoduls geführt werden. Das Funktionsmodul setzt dann die empfangenen Telegramme auf eine andere Gruppenadresse um. Es sendet sie aber nur auf die andere Gruppenadresse weiter, wenn die Funktion eingeschaltet ist.

Schaltung:



Ist das Signal 'Transfer frei' eingeschaltet und wurde gerade ein Telegramm auf das Signal 'Eingangssignal' empfangen ('C\_Eingangssignal' ist einen Taskzyklus lang auf 1), so hat der A-Ausgang des UND-Blockes kurz eine 1 am Ausgang. Dadurch schaltet der AASZ-Block den Wert des Eingangssignales auf das Signal 'Ausgangssignal'. Damit das Ausgangssignal immer gesendet wird, wenn ein Eingangssignal empfangen wird und die Funktion eingeschaltet ist, wird der A-Ausgang des UND-Blockes auf das Kontrollsignal von 'Ausgangssignal' geführt. Dies muss aber über den TRA-Block gemacht werden, da es nicht möglich ist, lokale Signale auf einen Ausgang zu zeichnen.

## 26 Betriebsstunden - Zähler

---

Diese Schaltung kann zum Zählen der Betriebsstunden z.B. von Lampen verwendet werden. Als Eingang dient ein binäres Signal, mit welchem der Zähler eingeschaltet (gestartet) oder ausgeschaltet (gestoppt) werden kann. Mit dem zusätzlichen Eingang "Reset" kann der Zähler wieder auf 0 zurückgestellt werden.

Nach Auftreten eines Bus-Resets oder eines Spannungsunterbruchs am Funktionsmodul, wird mit Hilfe des Speicher - Bausteines mit dem alten Zählerstand (Stand vor Bus-Reset) weitergefahren.

Die Schaltung arbeitet in einem zyklischen Task von einer Sekunde.

Der Betriebsstunden-Zähler ist mit zwei Stufen aufgebaut. In der ersten Stufe werden die Sekunden der angebrochenen Stunde und in der zweiten Stufe die Stunden gezählt. Erreicht der Sekundenzähler den Wert 3600 Sekunden (eine volle Stunde), so wird der Sekundenzähler auf 0 zurückgestellt und der Stundenzähler um 1 Stunde erhöht.

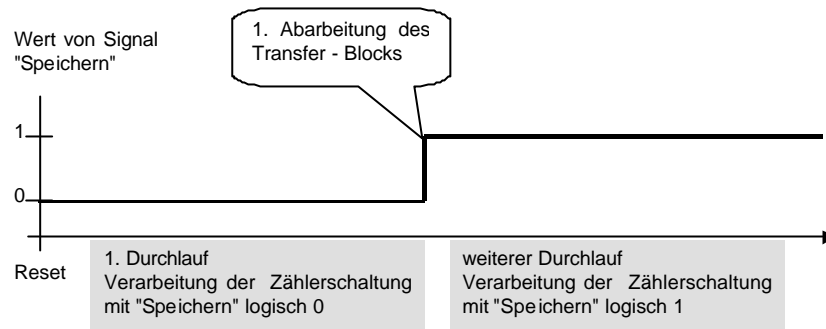
Der Aufbau der eigentlichen Zähler basiert auf der Rückkopplung des Zählersignals. Dabei wird zum Ausgangswert des Speicherbausteines mit einem Addierer ein Inkrementalwert hinzugezählt. Im Falle des Sekundenzählers wird dieser Inkrementalwert mit Hilfe eines EASZ Bausteines gebildet, welcher in Abhängigkeit des Start/Stop-Signals den Wert 0 oder 1 generiert. Beim Stundenzähler wird der Inkrementalwert auch mittels EASZ gebildet. Jedoch wird dieser EASZ bei einem Übertrag des Sekundenzählers angesteuert bzw. dann, wenn der Sekundenzähler den Wert 3600 erreicht.

Die Speicherung der Zählerstände wird durch das Signal "Speichern" gesteuert. Beim erstmaligen Abarbeiten der Schaltung z.B. nach einem Bus-Reset, darf dieses Signal nicht auf logisch 1 gesetzt sein. Ansonsten würden die gespeicherten Zählwerte von einem ungültigen Wert nach dem Bus-Reset überschrieben.

Das Signal "Speichern" wird deshalb mit dem Anfangswert 0 initialisiert. In der 4. Zeile des Segmentes wird dieses Signal durch einen Transferbaustein mit dem Wert 1 überschrieben. Der daraus resultierende Effekt erzielt folgendes zeitliches Verhalten:

## EIB System

### Funktionsmodul und FMTool



Beim ersten Durchlauf ist die Speicherung der Zählstände inaktiv. Mit der Verarbeitung des Transferbausteines wird die Speicherung aktiv gesetzt. In allen folgenden Verarbeitungszyklen werden danach die Zählstände gespeichert.

## 27 Index-Verzeichnis

---

### A

AASZ-Block 30, 31, 49, 59, 75

ADD - Block 76

ADD-Block 51

ALARM-Block 31, 49

Ändern von 'Konstanten' zur Laufzeit 56

Ausfallsicherheit 31

### D

Datentyp Value 14, 15

Datentyp-Umwandlungen 14, 15

DMFF-Block 17, 22, 24, 25, 31, 53

Doppelklick 53

### E

EAS-Block 56

EASZ-Block 8, 15, 43, 46, 51, 66, 71

Ein- und ausschalten von Funktionen 75

### F

Fallende Flanken 18

Fehler suchen 59

### G

GETD-Block 46, 66

GETT-Block 43, 62

### I

Initialisierungstask 71

### J

Jalousie automatisch steuern 69

Jalousie zeitabhängig steuern 62

### K

KLEIN-Block 12, 15, 66, 69

Konstanten 56

Kontroll-Signale 7

### O

Oder-Verknüpfung von Gruppenadressen 8

OSZI-Block 25

### P

PACBW-Block 14

PACD-Block 46, 66

PACS-Block 14

PACSV-Block 14

PACT-Block 43

## **EIB System**

### **Funktionsmodul und FMTool**

PACVS-Block 15

R

RSFF-Block 8, 25, 31, 69

S

SETD-Block 46

SETT-Block 43

SMFF-Block 19

SPEICH - Block 76

Steigende Flanken 17

SUHR-Block 62, 66

T

Tasten-Signale verarbeiten 19, 22, 25, 53

TEILD-Block 46, 66

TEILS-Block 15

TEILT-Block 43

TEILWB-Block 15

Telegramme verhindern 49

TFF-Block 24, 53

TRA-Block 7, 75

Tricks zum Debuggen 59

U

Uhr richten 43, 46

V

Vergleichen von Pulddauer 19

Verhindern von Telegrammen 30

VERZ-Block 31

W

WECHS-Block 18, 19, 31

Werte vergleichen 69

Z

Zähler 76

zeitabhängige Funktionsblöcke 71

Zeitwerte 71