

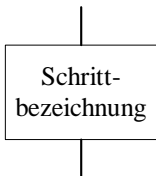


## Elemente der Ablaufsprache AS

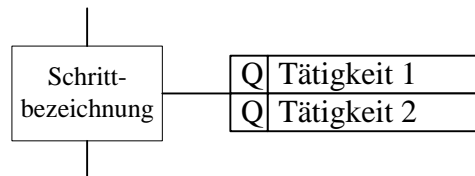
Eine Steuerung in der Ablaufsprache wird als ein Netz von Schritten und Transitionen dargestellt. Folgende Symbole sind festgelegt worden:

### Schritt:

IEC



erweitert



- Schrittbezeichnung:

Kurzbeschreibung des Schrittes, Schrittname

- Q: Qualifier

Der zu jeder Tätigkeit anzugebende Qualifier gibt an, wann die Tätigkeiten ausgeführt werden.

- Tätigkeiten:

Ansteuerung, Schaltbefehle oder ganze Steuerprogramme, die im entsprechenden Schritt ausgeführt werden.

### Grundschrift:



Der Grundschrift ist der Startschritt des Ablaufdiagramms.

### Transition Schaltbedingung:

mit



Die Schaltbedingung bestimmt, ob eine Transition schalten kann.

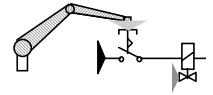
### Marke:



Marken dienen zur Simulation der Steuerung. Mit einer Marke bezeichnete Schritte sind aktiv, d.h. die zum Schritt gehörenden Tätigkeiten werden ausgeführt. Tätigkeiten nicht markierter Schritte, d.h. nicht aktiver Schritte werden nicht durchgeführt.



Zu Beginn der Simulation befindet sich im Quadrat des Grundschrilles eine Marke.



Eine Sequenz kann entweder mit 5 oder 20 Schritten gezeichnet werden, je nachdem, wie detailliert ein Ablauf beschrieben werden soll. Es ist wichtig, dass man den richtigen Grad der Detaillierung findet. In einem Extrem kann ein Diagramm aus mehr als 100 Schritten bestehen, jedes Detail ist in einem Schritt gelöst. Ein solches Diagramm ist nicht lesbar. Wird das gleiche mit 3 Schritten dargestellt, so ist das Diagramm bis zur Bedeutungslosigkeit verallgemeinert.

## Regel für das Zeichnen von Ablaufdiagrammen

Für die Erstellung von Ablaufdiagrammen gelten einige einfache Regeln. Werden diese eingehalten, so ist die Steuerung vollständig festgelegt, d.h. sie kann (könnte) automatisch aus Ablaufdiagrammen generiert werden.

### Regeln:

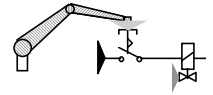
- Zwei Schritte können nicht direkt miteinander verbunden werden. Zwischen zwei Schritten liegt immer eine Schaltbedingung.
- Zwei Transitionen können nie direkt miteinander verbunden werden. Zwischen zwei Transitionen liegt immer ein Schritt.
- Jede Darstellung hat mindestens einen Grundschrift

Diese Regeln sind einfach einzuhalten indem leere Schritte, d.h. Schritte ohne Tätigkeit oder wahre Transitionen eingefügt werden.

Verbindungen sind immer mit einer Richtung versehen. Die Richtung zeigt die Bewegungsrichtung der Markierungen an. Diese ist in der Regel von oben nach unten. Für Verbindungen nach oben oder zur Verbesserung der Lesbarkeit, werden Pfeile auf den Verbindungslinien eingezeichnet.

## Anfangsmarkierung

Mit dem markierten Grundschrift wird der Ausgangszustand fest gelegt. In der Ablaufsprache nach IEC61131-3 gibt es oft nur einen einzigen Grundschrift. Wenn nun mehrere Prozesse parallel ablaufen, so ist dies an einem Start paralleler Aktionen nach dem Grundschrift erkennbar. Dies ist ein wesentlicher Unterschied zum Petri-Netz. Dort können Anfangsmarkierungen fast beliebig sein.

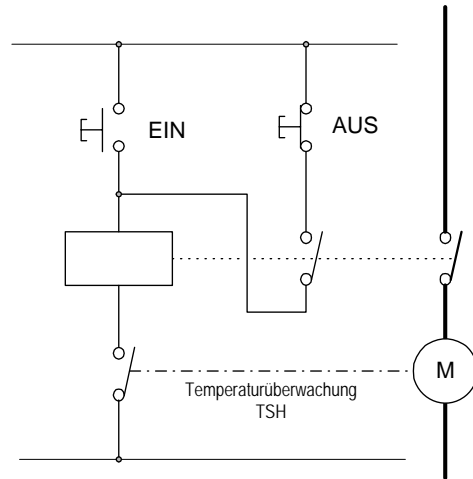


## Das Beispiel

Anhand des folgenden Beispiels soll die Ablaufsprache eingeführt. Ein Motor kann mit einem EIN-Taster eingeschaltet, mit einem AUS-Taster wieder ausgeschaltet werden. Eine Temperaturüberwachung verhindert eine Ueberhitzung des Motors. Die Relaisschaltung:

Zustände des Motors: läuft, steht

Schaltvorgänge: Relais zieht an, Relais fällt ab  
EIN-schalten, AUS-schalten



Ablaufdiagramm für die Motorensteuerung:

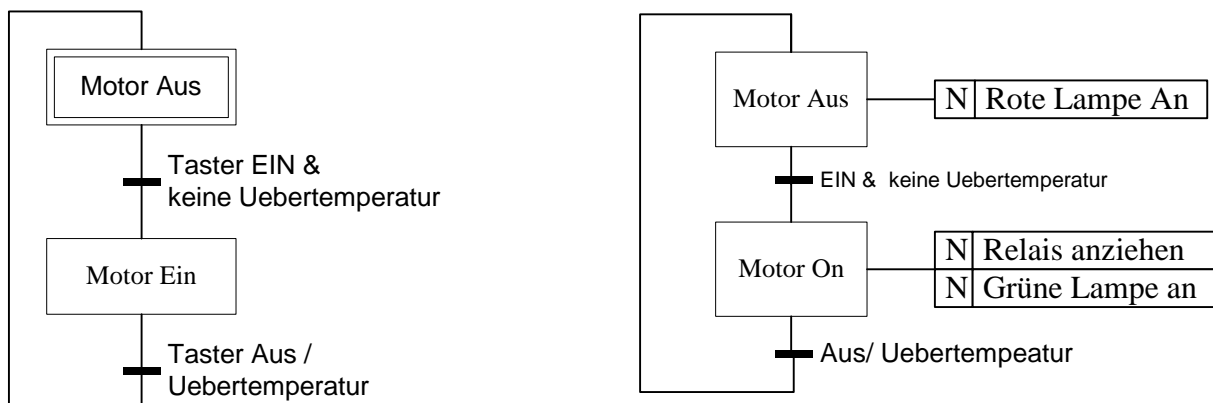
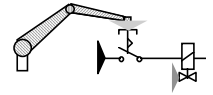


Abbildung: Motorenbeispiel

Das Diagramm links zeigt den Ablauf ohne Angabe von Aktionen. Rechts hingegen sind die Aktionen angegeben. Wie diese ausgeführt werden und was das ‚N‘ links der Transition bedeutet, ist Gegenstand des nächsten Abschnittes.

## Berechnung eines Automaten

Mit einem Graphen kann die Ablauflogik schön dargestellt werden, wie der Automat aber im Detail funktioniert, ist darin nicht ersichtlich. Insbesondere ist nicht klar, was der Automat 'macht', wenn ein paar Schritte markiert sind und keine Transition schalten kann oder was genau passiert, wenn eine Transition schaltet. Automaten werden heutzutage meistens auf einem Rechner implementiert, d.h. es muss ein Ablaufprogramm programmiert oder konfiguriert werden. Damit die Ablauflogik nun berechnet werden kann, muss das Ablaufprogramm entweder zyklisch aufgerufen werden oder über Strukturen verfügen, die eine ereignisorientierte Bearbeitung der Ablauflogik ermöglichen. Da in SPS-Systemen die Möglichkeiten zur ereignisorientierten Programmierung meistens fehlen (oft auch nicht



erwünscht sind), sollen im Folgenden die sehr wichtigen Details eines zyklisch berechneten Automaten genauer betrachtet werden.

## Ausführung von Aktionen

### Allgemeines

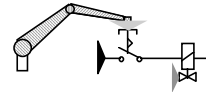
Die wesentlichen Unterschiede zwischen verschiedenen Automaten ergeben sich bei der Ausführung von Aktionen. Darum soll in diesem Abschnitt das Thema etwas weiter betrachtet werden, als dies für die Ablaufsprache eigentlich nötig wäre. Dieser Aufwand lohnt sich darum, weil meist diesbezügliche Unsicherheiten die Wurzel für Probleme beim Einsatz von Automaten bilden. Hinter dem Begriff 'Aktion' verbergen sich sehr unterschiedliche Aufgaben. Diese sollen zunächst in zwei Typen systematisiert werden.

#### einfache Aktion:

Eine einfache Aktion ist eine resultat-orientierte Kommunikations- oder Verarbeitungsaufgabe. Sie besteht in der Regel aus einer Folge von Meldungen oder Operationen, die übermittelt bzw. ausgeführt werden. Einfache Aktionen können innerhalb sehr kurzer Zeit erledigt werden. Sie werden bei einem Schrittwechsel einmal durchgeführt. Operationen erlauben, einfache 'Schaltungen' durchzuführen. Mit Meldungen können andere Prozesse gesteuert werden, d.h. beispielsweise gestartet oder gestoppt werden, die andere Automaten oder Steuer- und Regelaufgaben enthalten. Die Prozesse bestimmen aber selber, nach welchen zeitlichen Regeln diese bearbeitet werden sollen. Es ist klar, dass für die Realisierung selbständiger Aktionen entweder Mehrprozessorsysteme oder Multithreading-Betriebssysteme nötig sind.

#### periodische aufgerufene Aktionen:

Mit periodisch aufgerufenen Aktionen können Verarbeitungsaufgaben gelöst werden, die periodisch immer wieder abgearbeitet werden müssen. Dies sind beispielsweise Regel- und Ueberwachungsaufgaben oder auch untergeordnete Steueraufgaben. Wesentlich ist dabei, dass der Zyklus des Aufrufs durch den Aufruf des Automaten selber bestimmt wird. Nach dem Aufruf einer solchen Aktion werden die in der Aktion enthaltenen Operationen abgearbeitet und nach Vollendung wird mit den Rückgabewerten wieder zum Automaten zurückgekehrt. Periodisch aufgerufene Aktionen bilden also keine eigenen Prozesse.



Beispiele:

Typ	Aktion	Beschreibung
einfache Aktion	Licht einschalten	Das Licht wird einmal eingeschaltet und bleibt ohne weiteres hinzutun eingeschaltet, bis ein Befehl 'Licht ausschalten' das Licht wieder löscht
periodisch aufgerufene Aktion	Temperaturüberwachung <ul style="list-style-type: none"> <li><math>T &gt; T_{\max}</math></li> </ul>	Es wird geprüft, ob die Temperatur $T$ grösser als der Grenzwert $T_{\max}$ ist. Diese Funktion braucht sich nicht um ihre zeitliche Durchführung zu kümmern, diese wird vom Automaten entsprechend seiner Zykluszeit veranlasst.

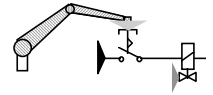
Tabelle 1: Beispiele für Ausführung von Aktionen

Ist eine Aufgabe komplex oder muss mit einer sehr kleinen Zykluszeit bearbeitet werden, so muss man sich folgende Strukturierungsfrage stellen: soll die Aufgabe als eine Aktion des Automaten oder als eigener, mit dem Automaten kommunizierender Prozess programmiert werden? Die beiden Varianten sind in der Tabelle 2 dargestellt.

<p>Ferngesteuerter, autonomer Regler:</p> <p>Der Regler wird mit einer Meldung (einfache Aktion) eingeschaltet und regelt selbständig, d.h. mit einer von der Berechnung des Ablaufautomaten unabhängigen Zykluszeit.</p>	
<p>Periodisch, durch Automaten aufgerufener Regler:</p> <p>Das Reglerprogramm wird durch den Automaten aufgerufen. Die Speicherung des Zustandes muss zwischen zwei Aufrufen gewährleistet sein.</p>	

Tabelle 2: Varianten für Ausführung von Aktionen

Überlegen Sie sich die Vor- und Nachteile der beiden Varianten!



### Parametrierbare Aktionen

Um die Zahl verschiedener Aktionen möglichst klein zu halten, kann es sinnvoll sein, neben dem Start und Stopp auch Parameter, beispielsweise Fahrweisen und Sollwerte, zu übergeben. Automaten, die in der Lage sind, schrittabhängig Parameterwerte an Aktionen zu übermitteln, werden als Rezeptsteuerungen bezeichnet. Das Ablaufdiagramm (die Ablauflogik) und die zu jedem Schritt und Aktion gehörenden Parameter bilden dann das Rezept.

### Aufruf der Aktionen

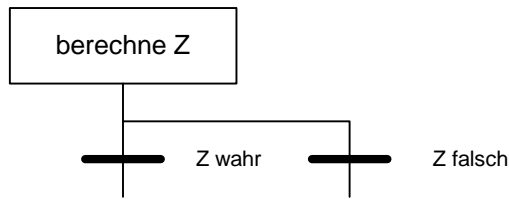
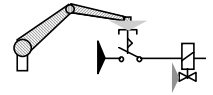
Die verschiedenen Arten von Aktionen kommen an unterschiedlichen Stellen in einer Ablaufdarstellung zum Einsatz. Folgende Stellen kommen dafür in Betracht: beim Schalten einer Transition, beim Erreichen eines neuen Schritts, nach dem Erreichen eines neuen Schritts und vor dem Verlassen eines Schritts. In der folgenden Tabelle sind die Anwendungen zusammengestellt:

	im Ablauf	Aktion
	beim Schalten einer Transition	einfache Aktion in der Ablaufsprache nicht möglich!
	beim Eintritt in einen Schritt, d.h. wenn der Schritt in der Berechnung des Ablaufs das erste Mal markiert ist.	einfache Aktion Initialisierung und Berechnung einer zyklischen Aktion
	Stationäre Markierung von Schritten.	Berechnung zyklischer Aktionen
	vor dem Verlassen eines Schrittes, d.h. wenn die Schaltbedingung bereits wahr ist.	einfache Aktion Berechnung und 'Abschalten' einer zyklischen Aktion

Tabelle 3: Ausführung von Aktionen

Zu Recht stellt man sich die Frage, warum man nicht einfach die einfachen Aktionen den Transitionen und die periodischen Aktionen den Schritten zuordnet. Unterschiede in der Zuordnung der Aktionen zu den Schritten und den Transitionen werden in den Automaten von Moore und Mealy deutlich. Im Mealy-Automaten sind Aktionen ausschliesslich mit Transitionen, im Moore-Automaten mit den Zuständen/Schritten verbunden. Dabei sehen beide nur einfache Aktionen vor. In der Ablaufsprache nach IEC61131-3 können Aktionen nur den Schritten zugeordnet werden. Dies hat folgende Vorteile:

- es sind direkte Verzweigungen im Ablauf möglich, die durch das Resultat einer Aktion bestimmt werden:

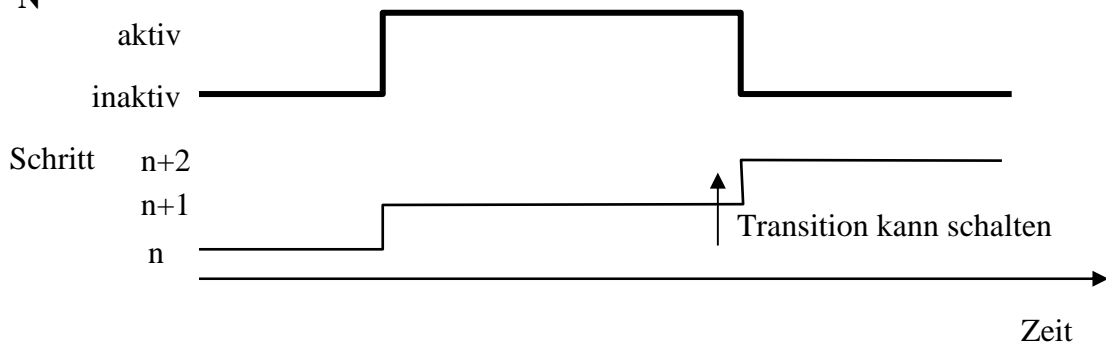


Falls Aktionen nur Transitionen zugeordnet werden können, ergeben sich zusätzlich Schritte. Zeichnen Sie eine mögliche Lösung!

Werden Aktionen ausschliesslich den Schritten zugewiesen, so stellt sich das Problem, wie die in Tabelle 3 zusammengestellten Fälle insbesondere, die einfache Aktionen realisiert werden. Um dieses Problem zu lösen, wurden die Aktionen mit Qualifier versehen. Diese bestimmen, wie die Aktionen in den Schritten ausgeführt werden.

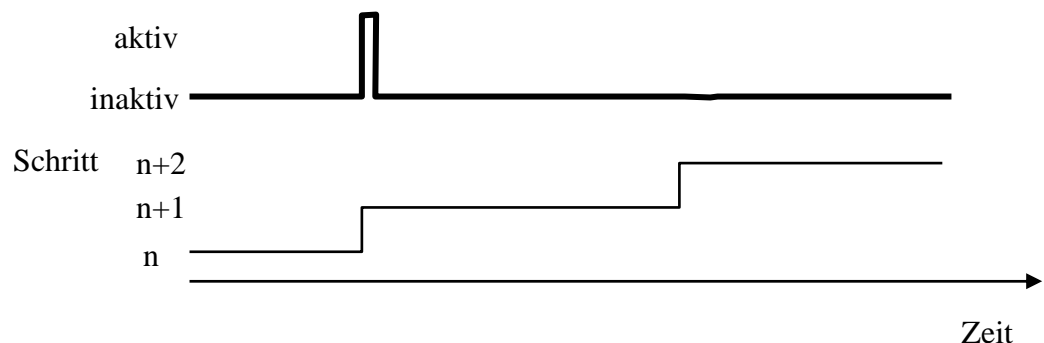
### Qualifier der Aktionen nach IEC61131-3

#### Qualifier 'N'

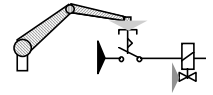
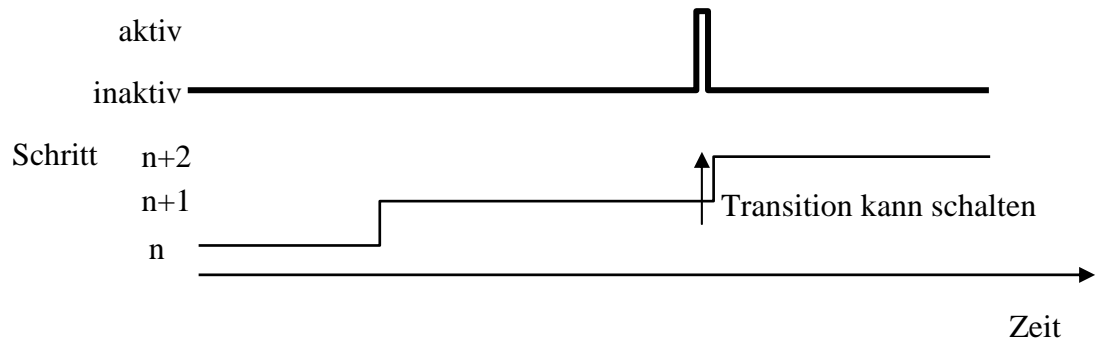


Die Aktion mit dem Qualifier N wird nach dem Schrittwechsel aktiv und wird berechnet, bis in den nächsten Schritt geschaltet wird. Nachdem die Schaltbedingung wahr ist und ein Schrittwechsel erfolgen kann, wird der Schritt ein letztes Mal durchgeführt. Dies ist nötig, damit Signale zurückgesetzt werden können, so z.B. ein Motor ausgeschaltet werden. Der Qualifier N eignet sich sehr gut für periodische Aktionen.

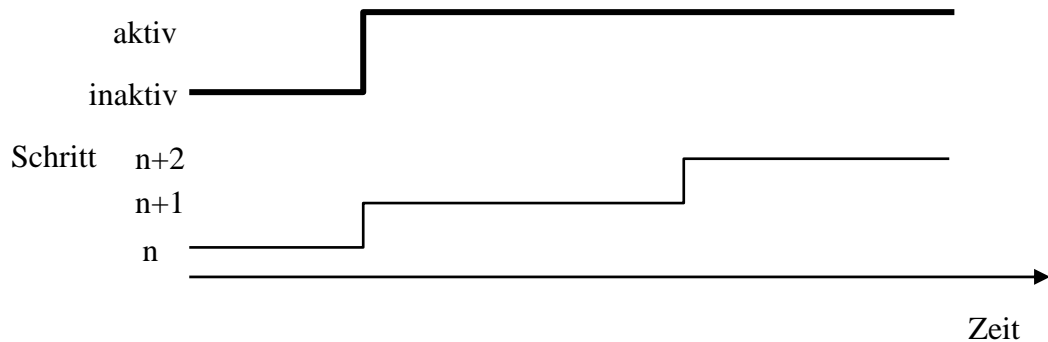
#### Qualifier 'P+'



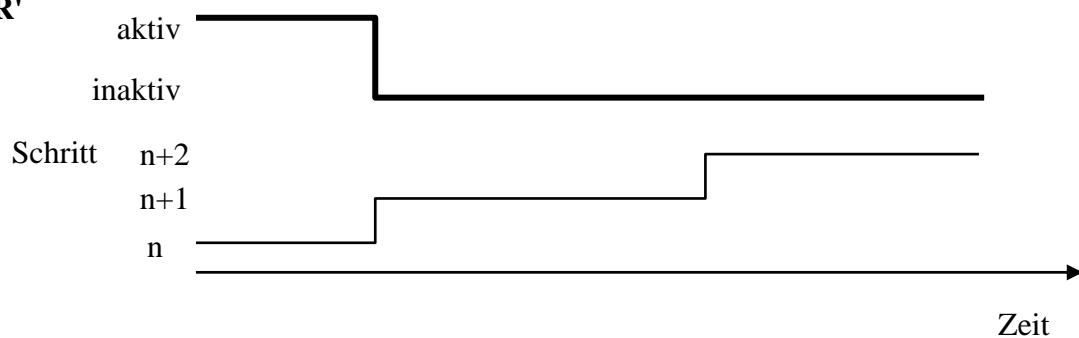
Eine Aktion des Schritts n+1 mit Qualifier P+ wird einmal nach dem Schrittwechsel von n auf n+1 durchgeführt.

**Qualifier 'P-'**

Eine Aktion des Schritts  $n+1$  mit Qualifier P- wird einmal vor dem Schrittwechsel von  $n+1$  auf  $n+2$  durchgeführt.

**Qualifier 'S'**

Eine Aktion des Schritts  $n+1$  mit Qualifier S wird immer nach dem Schrittwechsel von  $n$  auf  $n+1$  durchgeführt, bis die Aktion mit 'R' wieder gestoppt wird.

**Qualifier 'R'**

Eine Aktion des Schritts  $n+1$  mit Qualifier R wird immer nach dem Schrittwechsel von  $n$  auf  $n+1$  gestoppt.