



Handshake programmieren

Wir haben etwas eher ungewöhnliches vor : wir wollen mit einem PC Signale im Zeitverlauf verarbeiten. Üblicherweise startet man auf dem PC ein Programm, gibt vielleicht Eingaben ein, es rechnet, und terminiert dann wieder.

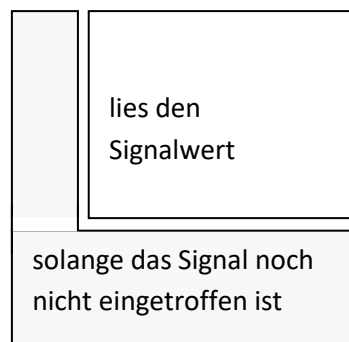
Anders die SPS : sie rechnet dauernd im Zyklus, und überprüft mit Bedingungen die Zustände von Eingangssignalen. Ein komplizierterer zeitlicher Ablauf wird oft mit Zustandsmaschinen realisiert, gesteuert von "Zustandsmerkern" oder ähnlichem.

Die Kernoperation dieser Aufgaben ist das Warten auf ein Signal, bei dessen Eintreffen irgendetwas ausgeführt wird.

Struktur : "aktives Warten (polling)"

Will man auf ein Ereignis warten, kann man bei der SPS einfach danach fragen (eine bedingte Ausführung, also "if"-Struktur). Das Programm kommt ja im Zyklus immer wieder daran vorbei. Die Programmlogik dahinter macht man mit Zustandsabläufen.

Beim PC kann man das so machen, daß die Warterei vom Anwender in seinem Programm angelegt wird. Also eine Schleife :



Dieses Verfahren wird in der Informatik als „polling“ (aktives Warten) bezeichnet. Es hat eine Reihe von Nachteilen. Die Warteschleife verbläst die komplette Rechenleistung (was manchmal aber nicht schadet, wenn nichts anderes zu tun ist). Außerdem könnte bei einem Fehler (wenn z.B. das erwartete Signal wegen einem Hardwareproblem nie kommt) der Rechner hängen bleiben (das kann man durch Prüfprogramme, sog. Watchdog-Timer, beheben)

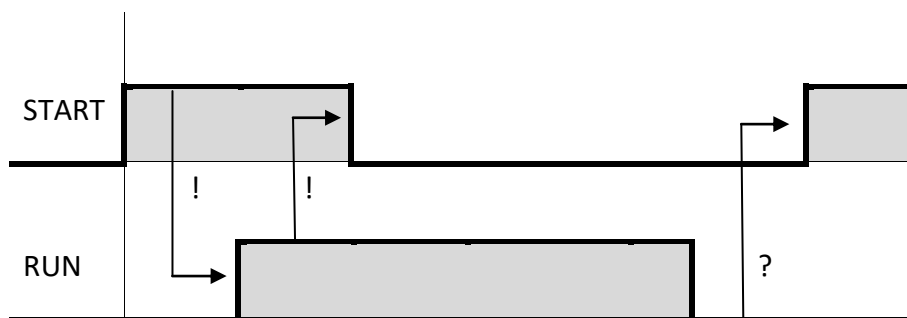
Struktur : "Zyklus mit Zustandsmaschine"

Man macht es einfach wie bei einer SPS, baut also um das Steuerprogramm einfach eine Endlosschleife außen rum. Dann programmiert man die Abläufe im Programm, wie man es von der SPS her gewohnt ist. Auch hier wird man für kompliziertere zeitliche Abläufe oft eine Zustandsmaschine benötigen, aber das kennen sie ja aus der Steuerungstechnik !

Das mag jetzt ein wenig übertrieben wirken, und hält uns vielleicht auch ein Wenig auf, aber es schadet nicht, über solche Strukturen mal grundlegend nachdenken.

Machen wir ein Beispiel :

Ein MES-System startet eine SPS, indem das Signal START auf 1 gesetzt wird. Als Handshakequittung soll ein Signal RUN=1 erfolgen. START wird wieder zurückgesetzt. Nun beginnt die Mechanikaktion an der SPS. Solange diese läuft, bleibt RUN=1 (Funktionshandshake). Erst nach RUN=0 darf erneut gestartet werden.



mit polling-Struktur :

Als I/O-Befehle benutze ich symbolisch "read" und "write", in der Realität sind das irgendwelche Methoden (z.B. ReadOPCTag).

```
START = 0

while True:

    r = read(RUN)
    while r == 1:
        r = read(RUN)

    write(START, 1)

    while r == 0:
        r = read(RUN)

    write(START, 0)
```

Der zeitliche Ablauf wird durch die polling-Schleifen geregelt, das Programm bleibt da jeweils stehen und wartet ...

(Bemerkung : das erste r=read(RUN) ist nötig, weil bei der Schleife sonst beim ersten Mal der Vergleichswert r unbekannt ist)

mit Zustandsmaschinen-Struktur :

Die Zustandsmaschine ist hier stark übertrieben, das geht auch einfacher, aber ich möchte die Struktur gut sichtbar machen :

```
state = 0
START = 0

while True:

    if state == 0:

        r = read(RUN)
        if r == 0:
            state = 1
            write(START,1)

    if state == 1:

        r = read(RUN)
        if r == 1:
            state = 0
            write(START,0)
```

Der zeitliche Ablauf wird durch die Zustandsmaschine geregelt.