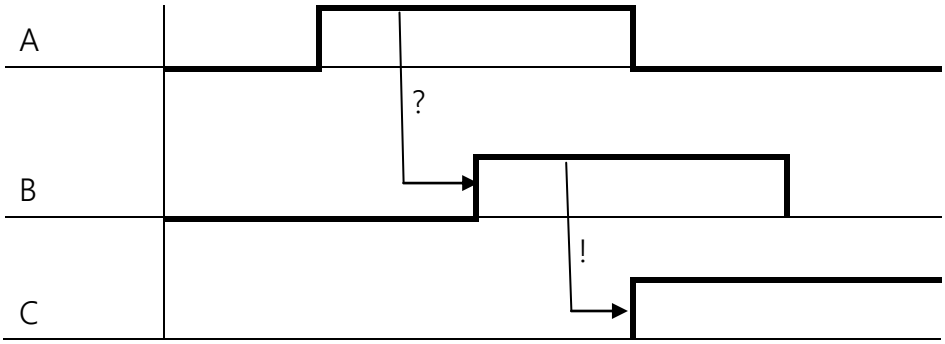
Handshake-Protokolle

In Industrieanwendungen wird bei der Kommunikation zwischen Geräten meist vom Handshakeprinzip Gebrauch gemacht. Das bedeutet im Wesentlichen, daß alle Signale vom Kommunikationspartner quittiert werden (wie beim Polizeifunk), und daß Signale immer abhängig vom Zustand des Partners erfolgen. Das minimalste Handshake-Protokol wäre ein quittiertes Start-Signal.

Timing-Diagramm

Zur Darstellung von digitalen Kommunikationsabläufen ist ein Timingdiagramm nützlich. Es zeigt den zeitlichen Ablauf qualitativ, also ohne Angabe von Einheiten. Über einer gemeinsamen Zeitachse werden die Signalspuren angetragen.

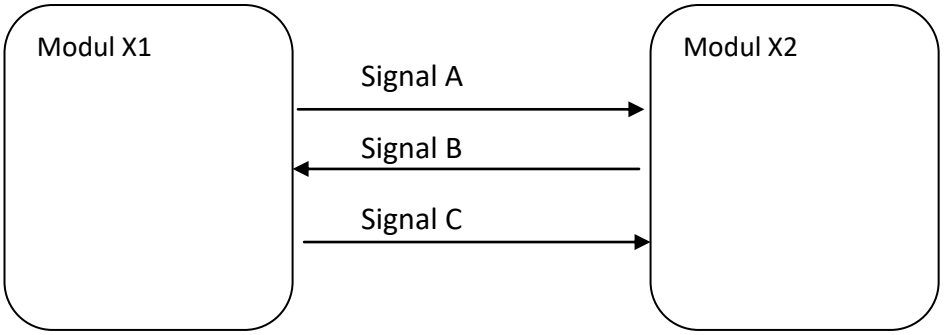
Wenn man will, kann man Kommentare dazusetzen. Oft wird das so gemacht : mit dem ? – Symbol werden Bedingungen definiert (das Signal kann nur erfolgen, wenn die Bedingung vorhanden ist, muß aber nicht), mit dem !-Symbol werden zwingende Abfolgen definiert (das Signal muß folgen).



Bitte beachten : ein Pfeil muß immer von einem Pegel ausgehen, nicht von einer Flanke, weil der exakte Flankenzeitpunkt (z.b. bei Bussystemen) oft nicht erkannt werden kann. Er führt aber zu einem Signalwechsel, und das ist immer eine Flanke.

Nützlich, auch wenn es trivial erscheint, ist eine kleine Skizze, die zeigt, welches Signal an welchem Modul Ein- und Ausgang ist.

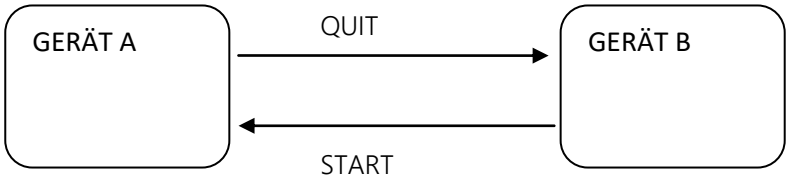
Diese Information ist im Timing-Diagramm nicht vorhanden.



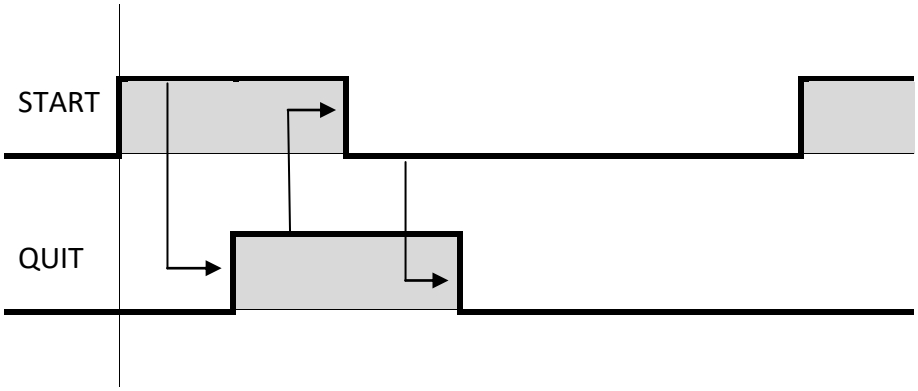
Beispiel für einen Handshake

Zwei Geräte (A und B) tauschen Nachrichten aus. Gerät B kann Gerät A durch das Signal START=1 starten. START und QUIT führen einen Handshake aus. (QUIT quittiert also START) QUIT wird = 1 sobald START erkannt ist (intern läuft jetzt z.b. eine Mechanikaktion in A an). Mit QUIT=1 wird START zurückgesetzt. (Die START-Info ist nicht mehr nötig, weil sie ja quittiert wurde) Mit erkanntem START=0 ist nun auch QUIT nicht mehr nötig, und wird zurückgesetzt. Der Ablauf beginnt von vorne.

Blockbild :



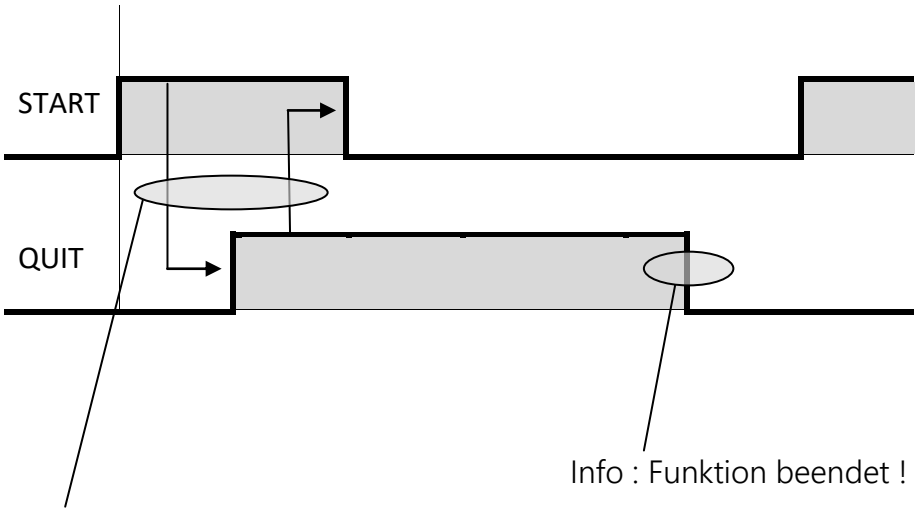
Timingdiagramm :



Im **Signalhandshake** führen die Quittungen immer sofort zur Rückname des auslösenden Signales, und die Quittung wird auch sofort wieder rückgesetzt, wenn das Signal reagiert hat, wie hier in diesem Bild.

Will man das Partnergerät nun auch noch darüber informieren, wann der gestartete Ablauf (z.b. eine längere Mechanikaktion) beendet ist, braucht man dazu noch ein weiteres Signal.

Man kann das aber auch durch einen **Funktionshandshake** kommunizieren. Dabei wird der Beginn des Quittungssignals zum Handshake benutzt wie bisher. Das Quittungssignal wird aber nicht wie oben sofort zurückgesetzt, sondern bleibt für die Funktionsdauer des Mechanikablaufs gesetzt. Rücksetzen der Quittung bedeutet hier : "Modulaktion beendet".



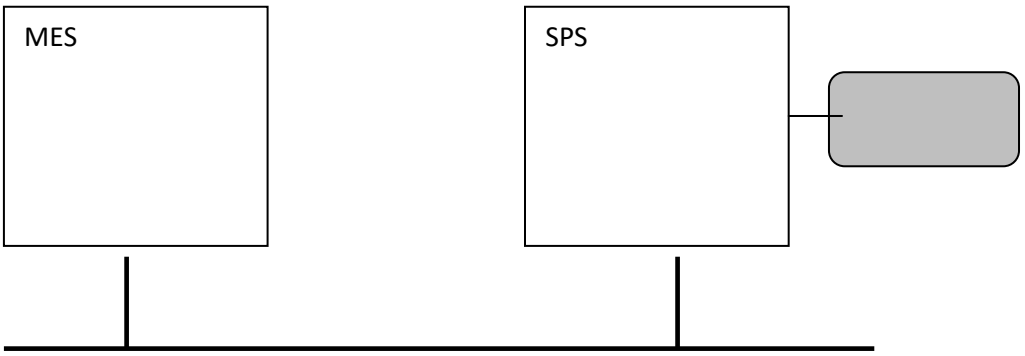
Ein Detail noch : Das Ende einer Aktion eines Moduls kann in "relevantes Funktionsende" und "absolutes Funktionsende" unterschieden werden. Das relevante Ende ist dann erreicht, wenn das Modul die anderen Module nicht mehr beeinflußt, wenn z.b. ein Teil von einem Roboter auf ein Transportband gelegt ist und der Greifer geöffnet hat. Das absolute Funktionsende ist dann erreicht, wenn das Geräte nichts mehr tut, also der Roboter z.b. in Referenzstellung (Ruhestellung) angelangt ist.

Welches davon nun für den Funktionshandshake benutzt wird, entscheidet der Entwickler.

Beispiel aus einer Technikerprüfung

Praxis

Blockbild :



Zwischen SPS und MES-System wird zur Kommunikation ein Handshake-verfahren eingesetzt :

Die **SPS** setzt bei Anlauf das Signal **READY .**
Will **MES** die SPS starten, wird zunächst mit einem Signal **TEST** in der SPS ein Selbsttest ausgelöst. TEST darf nur erfolgen, wenn READY = 1

Die **SPS** setzt daraufhin **READY** = 0. (Dies dient als HS-Quittung für TEST)

In der Mechanik des angeschlossenen Moduls läuft nun ein Selbsttest ab. Nach ca. 20 Sekunden ist dieser beendet, und die SPS setzt das Signal READY wieder 1.

(Ist der Selbsttest nicht erfolgreich, bleibt READY = 0, die SPS muß dann manuell rückgesetzt werden.)

Nach erkanntem Übergang von READY auf 0 und in der Folge wieder READY = 1 löst **MES** die SPS-Mechankfunktion mit **START** aus.

Im Sinne eines Funktions-Handshakes wird **READY** von der **SPS** zunächst = 0 und dann wieder = 1 gesetzt.

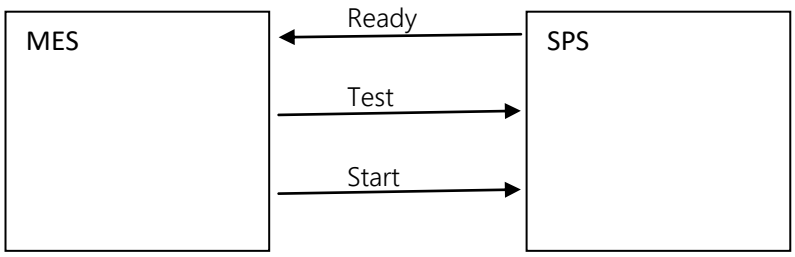
Aufgabe :

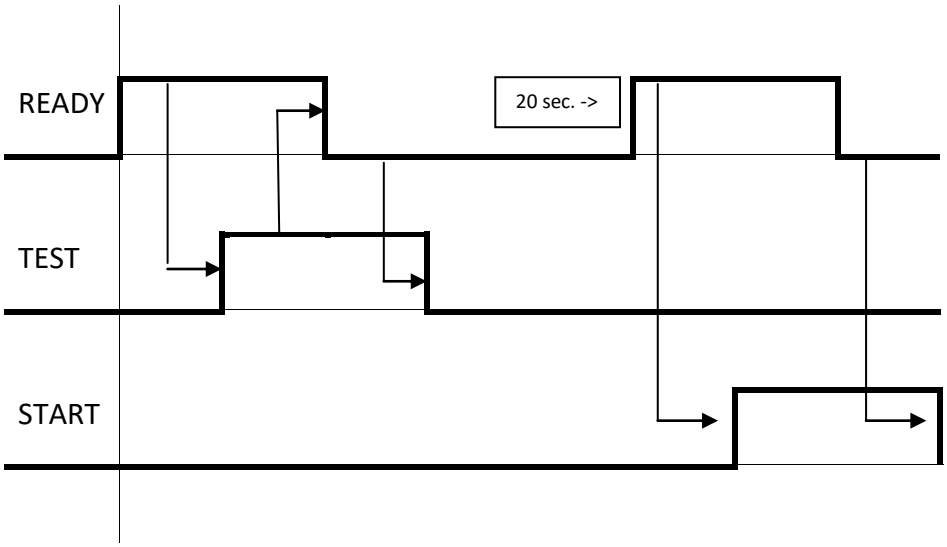
Zeichnen Sie ein Timing-Diagramm, das alle Kommunikationssignale der SPS enthält (erfolgreicher Selbsttest wird angenommen).

Zusatzinfo (Zeit abgelaufen, Mechanik fertig usw. dürfen nicht als Kommunikationssignale dargestellt werden ! Sie können solche Info im Sinne eines Kommentars aber dazuschreiben.

Lösungsvorschlag :

Zuerst das Blockbild. Das erscheint trivial, aber ich weiß, daß man hier Fehler machen kann mit den Signalrichtungen : dann ist der Rest Schrott !





Ready wird nach Aktion wieder 1 (Funktionshandshake), das passt oben aber nicht mehr rein ;-)