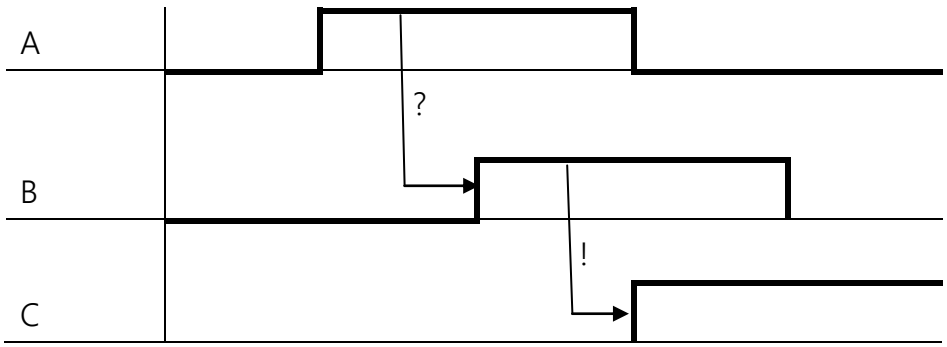
Handshake-Protokolle

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

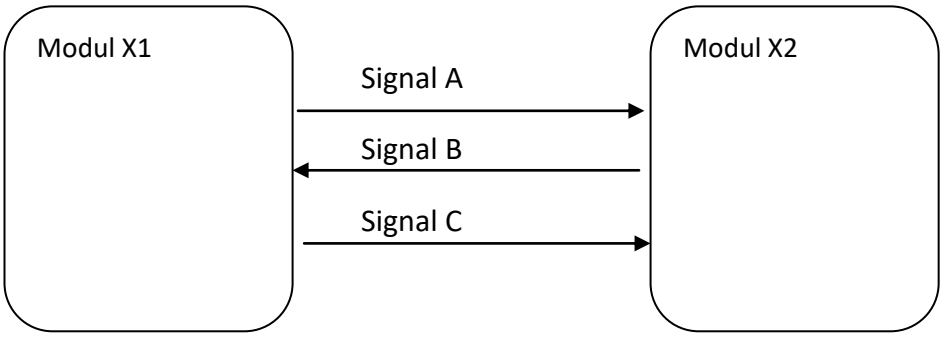
Zur Darstellung von digitalen Kommunikationsabläufen ist ein Timingdiagramm nützlich. Es zeigt den zeitlichen Ablauf qualitativ, also ohne Angabe von Einheiten. Über einer gemeinsamen Zeitachse werden die Signalspuren angetragen.

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).



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.



aktives Warten (polling)

Da die Arbeitsweise des Betriebssystems eines PC anders als die in der SPS ,muß man bei der Programmierung hier anders vorgehen Eine SPS kann mit einer Abfage (“if”) einfach prüfen, ob ein Signal eingetroffen ist. Der Signalpegel wird dann im Zyklus immer wieder abgefragt. Ein PC kann das zunächst so nicht realisieren : es fehlt der Ablaufzyklus.

Wenn man das Programm auf dem PC nicht auch zyklisch laufen läßt, muß der PC auf das Signal des Kommunikationspartners warten, indem er die Frage nach dem erwarteten Signal (hier z.B. START) immer wieder neu stellt. Also eine Schleife, die der Programmierer selbst einbauen muß.

Dieses Verfahren wird in der Informatik als „polling“ (aktives Warten) bezeichnet.

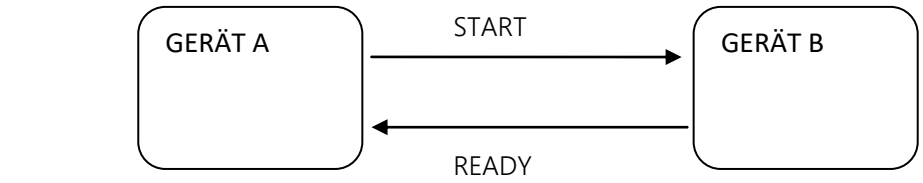
kleines Bilderrätsel zum Thema :

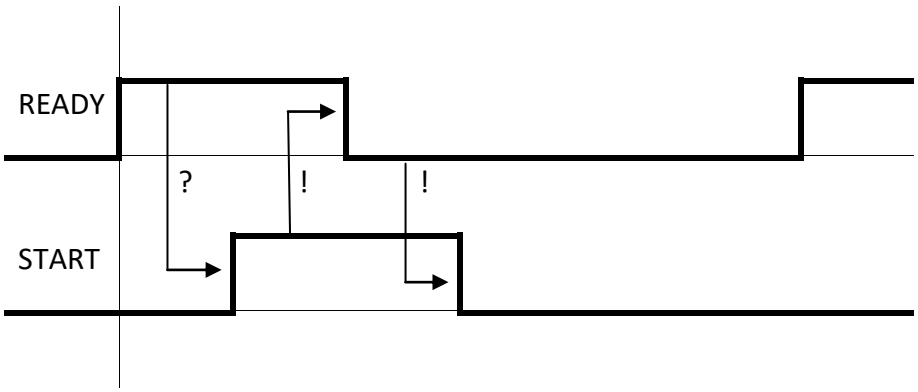
was ist hier falsch ?

Beispiel für einen Handshake

Zwei Geräte (A und B) tauschen Nachrichten aus. Gerät A meldet durch das Signal READY = 1 seine Betriebsbereitschft (z.b. nach Selbsttest). Gerät B kann (nicht "muß") Gerät A durch das Signal START=1 starten, wenn A betriebsbereit. START und READY führen nun einen Handshake aus. (READY quittiert also START) READY wird = 0 sobald START erkannt ist (intern läuft jetzt z.b. eine Mechanikaktion in A an). Mit READY = 0 wird START zurückgesetzt. (Der START ist nicht mehr nötig, weil er ja quittiert wurde !) A läuft nun bis zum Funktionsende seiner Mechanik, und setzt dann READY wieder=1 Der Ablauf beginnt von vorne ...

Blockbild und Timing dieses Ablaufes





Schreiben Sie eine Routine, die Gerät B realisiert !

1.) Entwerfen Sie zunächst (optional) ein Struktogramm.

2) Codieren Sie in Python.