| Fachschule für Elektrotechnik, Maschinenbautechnik und Metallbautechnik<br>der Landeshauptstadt München | |
|---|---|
| Klasse : | Name : MUSTER |

# Technikerprüfung 2012

# Automatisierungstechnik

# Zeit : 150 Minuten

| | Punkte: | Note : | Unterschrift: |
|---|---|---|---|
| Erstkorrektur | | | |
| Zweitkorrektur | | | |

# Teil 1 , ohne Unterlagen

Name, Klasse :

1.1 Welche Aussagen sind richtig ?

- O Dynamische Redundanz ist sicherer als statische Redundanz
- O TMR-Betrieb benötigt mindestens 2 identische, redundante Komponenten
- O “fail-save“ bedeutet, daß eine Komponente nicht ausfallen kann
- O Durch “burn-in“ wird die Temperaturbeständigkeit einer Komponente erhöht
- O vorbeugende Wartung verlängert den Bereich der Zufallsausfälle in der “Badewannenkurve“

1.2 Was versteht man unter einem Webservice ?

……

……

……

……

1.3 Ist die Kommunikation in Profinet RT deterministisch? Begründung !

……

……

……

……

1.4 Folgende Frage ist in einem Simatic-Forum aufgetaucht :

„Ich möchte meine S7 300 über das Profinet RT – Protokoll konfigurieren, also die Station laden. Geht das, oder muß ich dazu eine andere Hardware oder ein anderes Protokoll verwenden ?“

Was antworten Sie ?

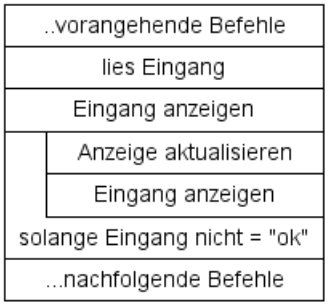
……

……

……

……

1.5 a) Geben Sie den Fehler in nachfolgendem Programmstück an, das auf das Eintreffen eines Signals „ok“ wartet :



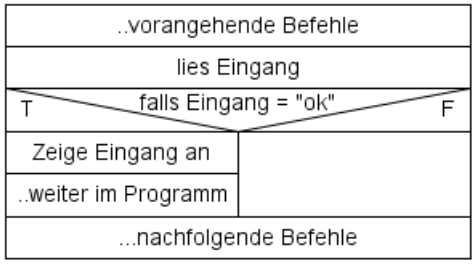

……

……

……

……

b) Was halten Sie von dieser Modifikation ? Begründung !




……

……

……

……

1.6 Welche Aussagen sind richtig ?

- O XML ist ein Übertragungsprotokoll, das Port 80 benutzt
- O Ein ERP-Server kann schnell auf Ereignisse im Maschinenpark reagieren
- O flow planing ist der Entwurf des Fertigungsablaufs (z.b. mit work plan)
- O Sequenzieren im MES bedeutet, die Produktreihenfolge zu ändern
- O Durch das http-Protokoll kann eine SPS sehr einfach mit MES kommunizieren
- O In Echtzeitkritischen Anlagen wird ERP am Feldbus (z.b. Profibus) angeschlossen

1.7 Welche Aussagen sind richtig ?

- O In Profibus DP ist keine direkte Master-Slave Kommunikation möglich
- O Profibus DP ist nur mit Master-SPS funktionsfähig
- O In Ethernet mit TCP/IP muß stets ein Gerät als Master betrieben werden
- O Profibus benötigt zur Signalübertragung nur eine zweipolige Twisted-Pair Leitung

1.8 Ein PC muß das Eintreffen eines Signals von einem Partnergerät durch aktives Warten (polling) ermitteln. Wieso muß das in einer SPS nicht so programmiert werden ?

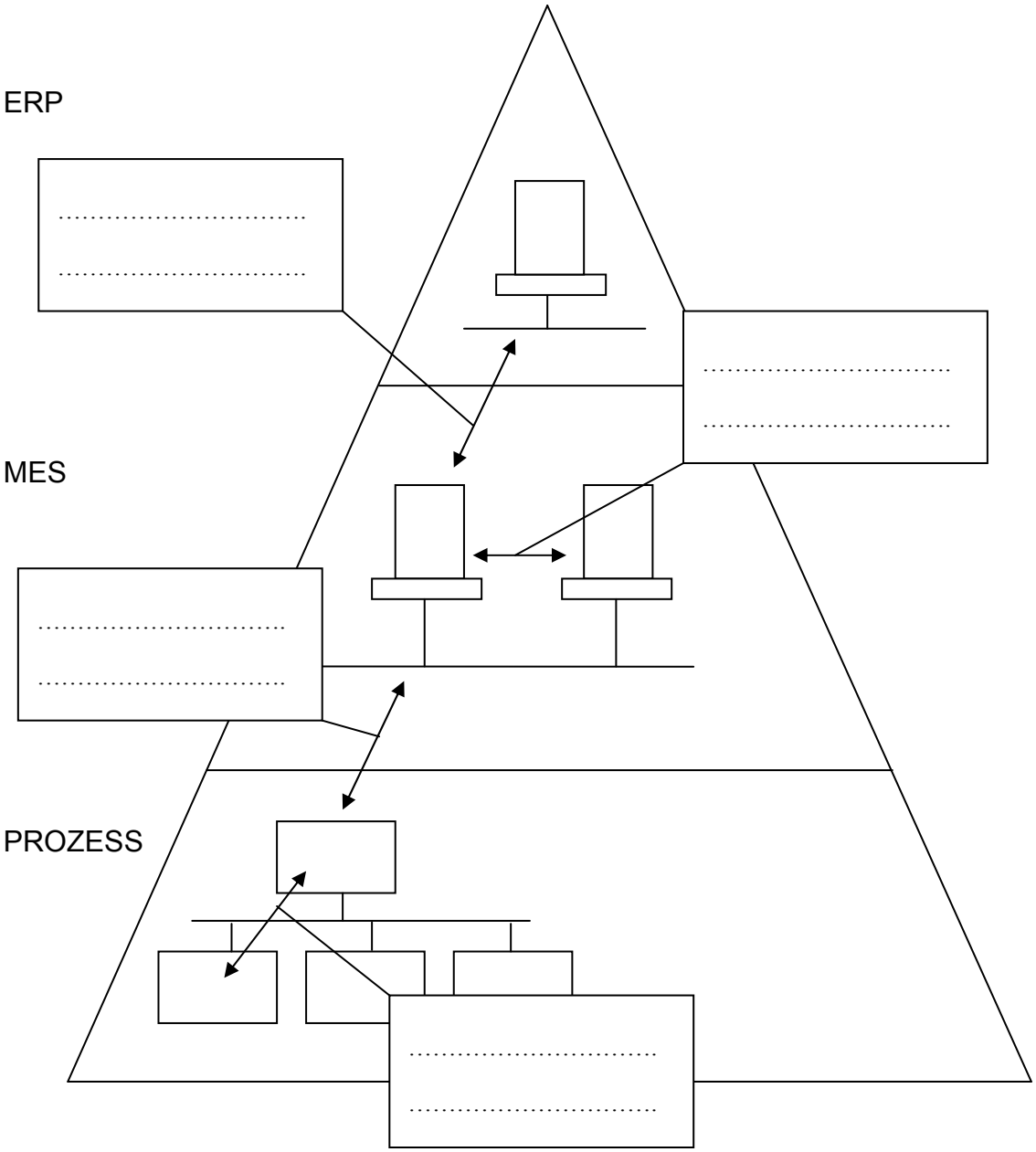
.....

.....

.....

.....

1.9 Geben Sie für die verschiedenen Kommunikationspfade (Pfeile) jeweils mindestens ein geeignetes Protokoll an :

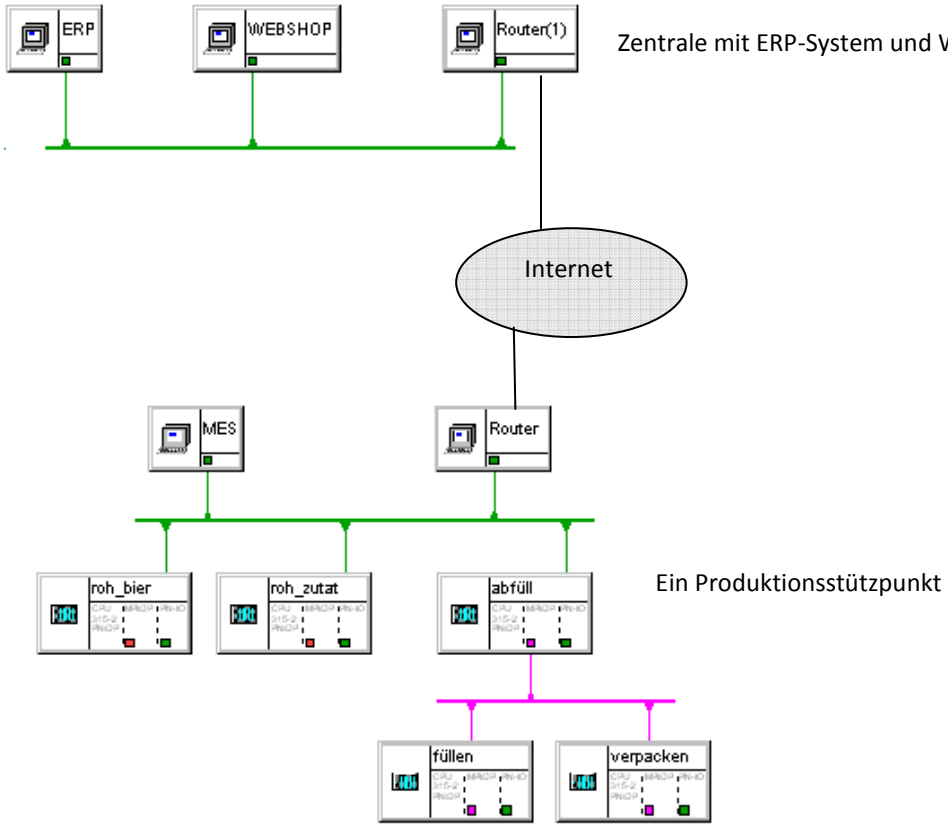


# Anlagendokumentation , gesamte Anlagenstruktur :

Das Internet-Unternehmen mybeer.com vertreibt weltweit Biermischgetränke. Das Geschäftsmodell sieht so aus, daß zentral am Firmensitz in Bad Tölz (BRD) die komplette Verwaltung usw. angesiedelt ist, und in den Vertriebsländern in allen größeren Städten jeweils kleine Produktionsanlagen (vollautomatisch, ohne qualifizertes Personal) unterhalten werden.

In den Produktionsanlagen wird, gesteuert von einem kleinen MES-System, eine einfache Biersorte hergestellt, die dann mit einer aus 25 wählbaren Zutaten in wählbarem Verhältnis gemischt und in Dosen mit vom Kunden selbst gestaltetem Etikett verpackt wird. Die Rezeptur und das Etikettdesign kann der Kunde im Webshop bei der Bestellung eingeben, am nächsten Tag holt er seine Bestellung (mindestens 100 Dosen) am Stützpunkt ab.

Hier sehen Sie die IT- und Anlagenstruktur dieser Firma :

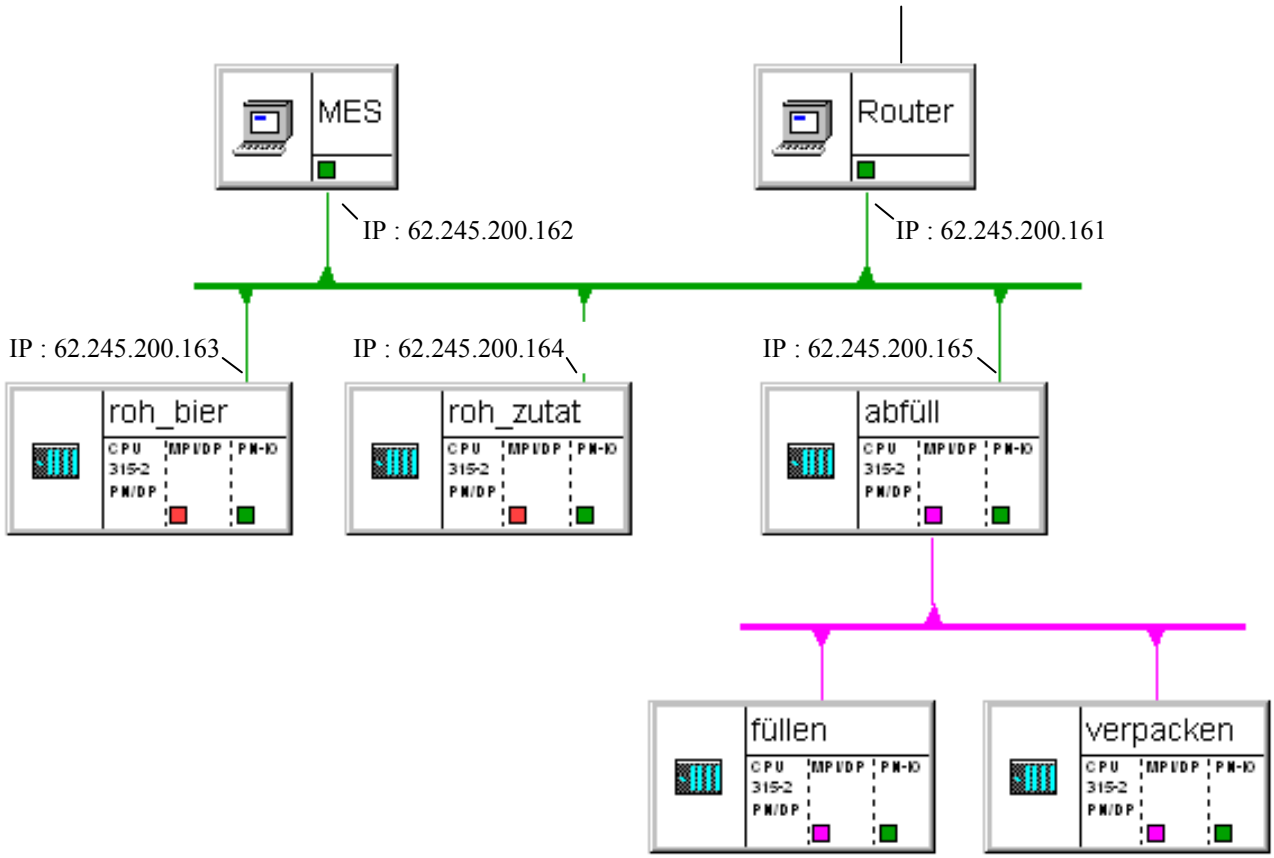


# Anlagendokumentation, Aufbau der Produktionsstützpunkte:

An den Produktionsstätten ist überall die gleiche Anlage aufgebaut :

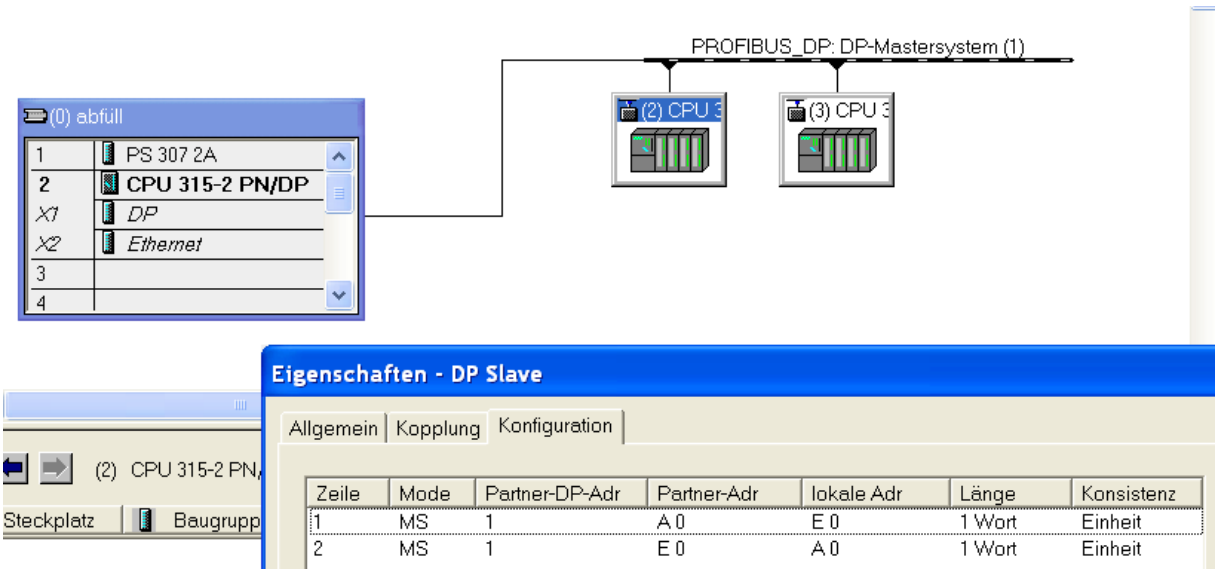
Ein MES-System unter Windows2008 steuert über TCP/IP drei angeschlosene SPS, die die Bierherstellung **roh_bier**, die Zutatenbereitung **roh_zutat** und die Abfüllung der Fertigprodukte **abfüll** übernehmen.

Wegen der zeitkritischen Vorgänge beim Füllen und Verschließen der Dosen sind diese beiden Teilprozesse als Slave-Module **füllen** und **verpacken** an der Master-SPS **abfüll** über Profibus DP gekoppelt.

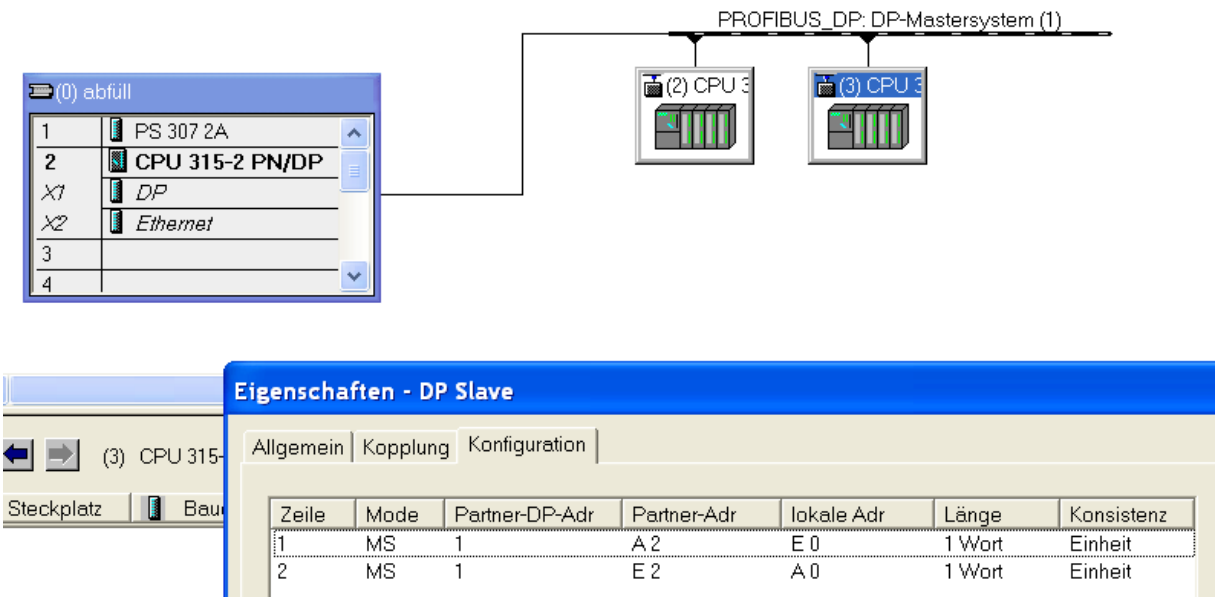


# Anlagendokumentation, Profibuskonfiguration :

## Anschluß der Slave-SPS füllen am Master abfüll :




## Anschluß der Slave-SPS verpacken am Master abfüll :

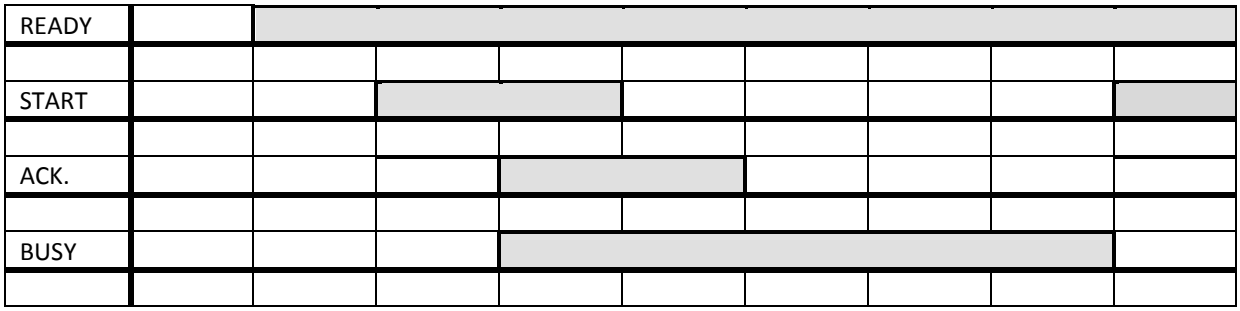


# Teil 2, mit Unterlagen

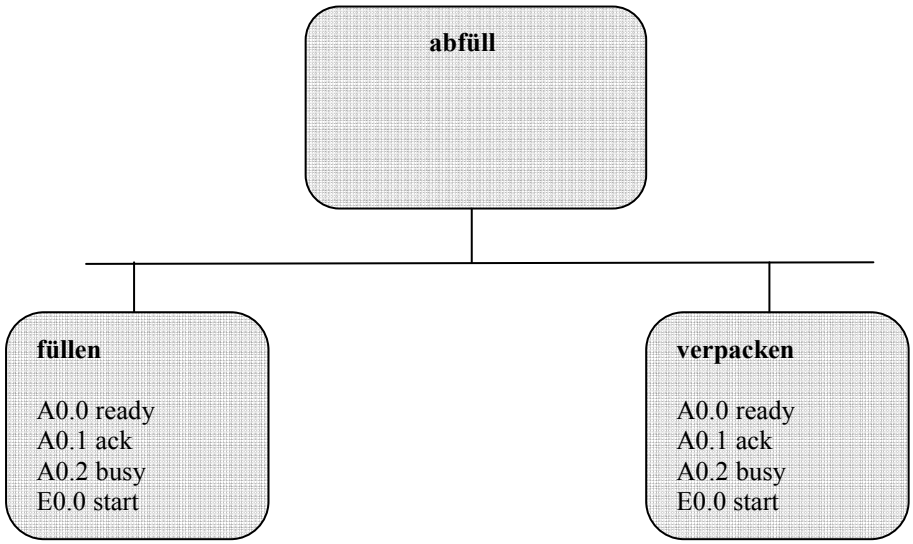
**Prozessebene :**

- **roh_zutat** und **roh_bier** füllen ihre Produkte in einen Zwischentank.
- **füllen** steuert den Abfüllvorgang in Dosen (aus dem Zwischentank).
- **verpacken** steuert das Handling der Dosen und deren Verpackung.

**füllen, verpacken** und **abfüll** folgen dem selben Kommunikationsprotokoll wie die SPS der digitalen Fabrik der Technikerschule :




Die Belegung der Kommunikationssignale in den Slaves ist wie hier angegeben festgelegt, die Zuordnung zu Adressen im Master ist den Simatic-Screenshots auf den Vorseiten zu entnehmen.



Aufgaben :

1. Ready von **füllen** und **verpacken** liegen in den SPS lokal an DB10.DBX1.1 an. Beide Ready-Signale sollen an beiden Slave-SPS an je zwei Meldeleuchten angezeigt werden, Ready(füllen) an **A124.3**, Ready(verpacken) an **A124.4** Hierzu stehen in den SPS für die Readysignale der anderen SPS die Eingänge **PartnerReady** bereit.

   Geben Sie alle nötigen Programmstücke an, um dies zu realisieren.

   Geben Sie bei jedem Programmstück unbedingt an, in welcher SPS dies ausgeführt wird !

2. Die Steuerung von **füllen** und **verpacken** soll in **abfüll** zyklisch durch starre, parallele Kopplung durchgeführt werden.

   Geben Sie als Petrinetz die Programmlogik an, die dies in **abfüll** realisiert. Eine ausgefallene Station (Ready=0) bewirkt einen Abbruch des Zyklus.

Für die nächste Teilaufgabe wird nun angenommen, daß :
a) **roh_bier** und **roh_zutat** direkt über TCP/IP miteinander kommunizieren
b) **abfüll** und **verpacken** über Profibus DP kommunizieren

3. Skizzieren Sie die beiden Kommunikationspfade a) und b) als ISO/OSI-Systeme.

Nun bauen wir für die nächste Aufgabe um :
**roh_bier** und **roh_zutat** sollen nun über Profinet IRT kommunizieren :

4. Erklären Sie mit einem kurzen Text, wie der Determinismus in Profinet IRT prinzipiell erreicht wird.

5. In einigen Profinet-Varianten wird die Master/Slave-Struktur aus Profibus DP beibehalten. Hierzu 2 Fragen :

   a) Ist dies technisch nötig ? (Begründung)
   b) Was ist der Grund für diese Struktur in Profinet ?

Nun ein letzter Umbau : wir lassen **roh_bier** und **roh_zutat** über Profinet CBA kommunizieren.

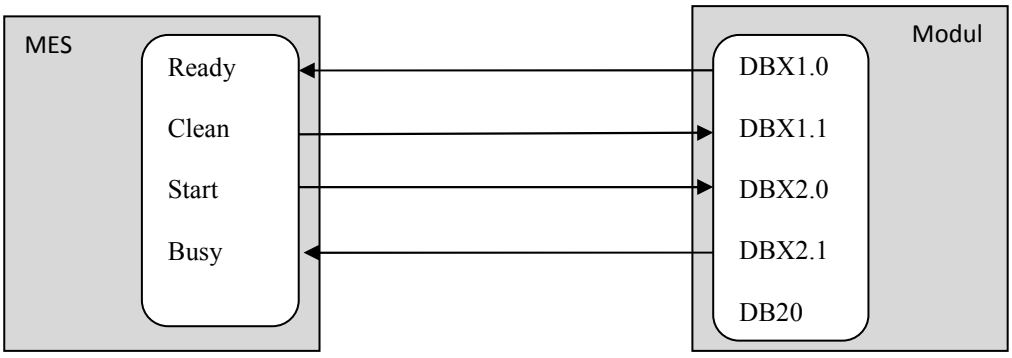
6. Jetzt läuft als Kommunikationskern DCOM. Erklären sie kurz, was hinter dieser Bezeichnung steckt (keine reine Übersetzung der Abkürzung bitte !)

**MES-Ebene :**

Die beiden Module zur Bereitung der Rohstoffe werden von MES mit einem einfachen Start/Busy-Handshake gesteuert.
Mit READY zeigt ein Modul nach dem Einschalten seine Betriebsbereitschaft.
MES setzt START, das Modul antwortet mit BUSY, welches sowohl als Handshake-Quittung für START (setzt START zurück) als auch als Funktionsdaueranzeige (bleibt 1, solange das Modul arbeitet) dient.

Weiter kann MES ein Signal CLEAN an ein Modul absetzen. Die Anzeige der Betriebsbereitschaft des Moduls READY geht damit auf 0, dies quittiert als Handshake CLEAN. Das Modul beginnt, alle Rohrleitungen rückzuspülen. Wenn der Vorgang abgeschlossen ist, geht READY wieder auf 1.

START oder CLEAN können nur erfolgen, wenn READY auf 1 und BUSY auf 0 steht.




7. Zeichnen Sie ein Timing-Diagramm, das alle Signale zwischen MES und einem Modul darstellt. Mit Pfeilen tragen Sie bitte alle (!) Signalbeziehungen ein, die auch zeigen sollen, ob ein Signal zwingend auf ein anders folgt, oder dies nur zur Bedingung hat.

In der SPS **roh_bier** werden die Kommunikation und die beiden Prozesse „bierherstellen“ und „rückspülen“ in verschiedenen Funktionsbausteinen realisiert.

Die Verbindung des Kommunikationsbausteins mit den Prozessbausteinen wird durch zwei „action“-Signale hergestellt.

**action_bier** startet den Prozessbaustein, der Bier herstellt.
Dieses meldet dann durch **/action_bier** (wird rückgesetzt), daß der Vorgang abgeschlossen ist**.**

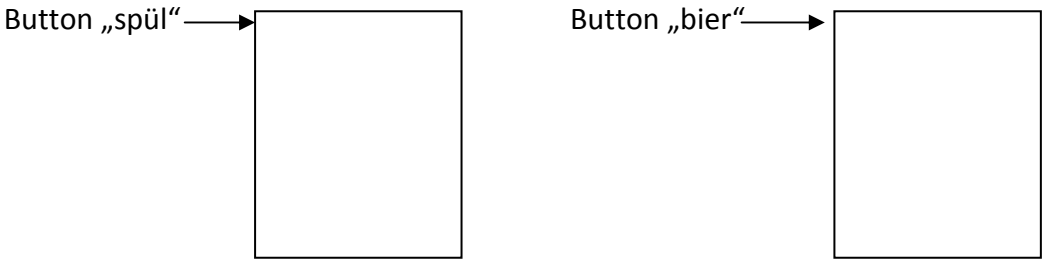
**action_clean** startet den Prozessbaustein, der rückspült.
Dieses meldet dann durch **/action_clean,** daß der Vorgang abgeschlossen ist**.**

Ein gleichzeitiges oder überlappendes Auftreten der beiden Steuersignale CLEAN und START ist durch die Programmlogik in MES ausgeschlosen und muß nicht beachtet werden.

8. Geben Sie mit einem Petrinetz einen Entwurf für den Kommunikations-FB in der SPS **roh_bier** an.

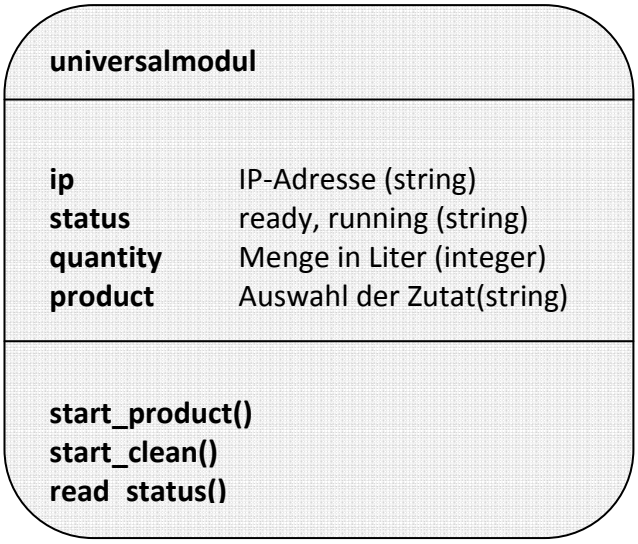
In MES wird die Behandlung der Peripherieschnittstellen später objektorientiert abgewickelt. Zunächst soll aber zu Testzwecken die Steuerung manuell ohne Objektorientierung realisiert werden. Mit Buttons (Symbolik im Struktogramm siehe unten) sollen die beiden Vorgänge vom Bediener startbar sein, falls dies nicht gelingt (z.b. nicht READY), müssen geeignete Fehlermeldungen erfolgen (Textbox), während des Vorgangs soll eine Zustandsmeldung (selbe Textbox) „running“ anzeigen und nach Fertigstellung auf „done“ wechseln.

Anmerkung : Sie dürfen nicht davon ausgehen, daß die Reaktion der SPS auf den CLEAN-Befehl unmittelbar erfolgt, sondern müssen hier, bedingt durch die Kommunikation, mit einer Verzögerungszeit rechnen.




9. Geben Sie als Struktogramm die Routine an, die auf MES-Seite den Vorgang „rückspülen“ wie oben beschrieben steuert.

Im Weiteren steht zur Ansteuerung der SPS-Module in MES eine Klassenbibliothek **beer.dll** zur Verfügung. Diese beinhaltet u.a. die Klasse **universalmodul.** Diese beinhaltet Attribute und Methoden für alle Module, wobei nur jeweils die sinnvoll benötigten mit Werten belegt werden.




**ip** ist die IP-Adresse der SPS (z.b. : “196.246.245.34“)

**status** meldet den Betriebszustand der SPS :
ready : Modul ist funktionsbereit
running: Modul arbeitet gerade (Spülen oder Produzieren)

**product** gibt an, welche Zutat in den Mischtank gefüllt werden soll.
(nur bei SPS roh_zutat belegt)

**quantity** gibt an, welche Menge der Zutat in den Mischtank gefüllt werden soll (nur bei SPS roh_bier und roh_zutat belegt)

**start_product()** ist die Methode zur Beauftragung eines Produktionslaufs

**start_clean()** ist die Methode zur Beauftragung eines Rückspülvorgangs
(nur bei roh_bier und roh_zutat sinnvoll)

**read_status()** ist die Methode zum Auslesen des Betriebsstatus

10. Geben Sie ein Struktogramm für ein Testprogramm an, das zunächst einen Spülvorgang in roh_bier durchführt , dann 150 liter Bier und 12 liter der Zutat “lemon“ parallel starr gekoppelt in den Mischtank befüllt, und dann einen Abfüllvorgang auslöst (nur auslösen, nicht bis zum Ende warten).

11. Codieren Sie das Programm aus Aufgabe 10 in VB .net (.. wie in Visual Basic 2010). Der Programmkopf (die Buttonbedienung usw.) ist nicht verlangt, die für die Nutzung der Klassenbibliothek nötigen Anweisungen geben Sie aber bitte mit an !

**ERP-Ebene :**

Die Kommunikation eines Standorts mit der Zentrale in Atlanta geschieht über das Internet mittels Webservices. Hierbei wird periodisch aus VB .net vom MES-System ein http-Request an ERP gestellt, den dieser mit einem XML-File beantwortet. Darin steht, ob für den Standort ein noch nicht bearbeiteter Auftrag ansteht, und seine Daten.

Folgende Informationen sind im XML-File enthalten :

| | |
|---|---|
| Auftrag : | ja oder nein (bei nein sind alle folgenden Angaben = 0) |
| Biermenge : | Menge in Liter |
| Zutat : | Name der Zutat |
| Zutatmenge : | Menge der Zutat in Liter |
| Etikett : | Name des .jgp-files (z.b. „meier20.jpg“) |

12. Geben Sie ein XML-File an, das diese Informationen übertragen kann. (Ein Beispiel mit Auftrag=ja bitte)