

Fachschule für Elektrotechnik, Maschinenbautechnik und Metallbautechnik  
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 ?

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

## 1.2 Was versteht man unter einem Webservice ?

Ein Dienst, der über eine URL angesprochen wird, und im http-Protokoll XML-Daten zurückliefert

## 1.3 Ist die Kommunikation in Profinet I/O deterministisch? Warum ?

Nein, weil Profinet I/O auf Ethernet und TCP/IP (oder UDP) basiert.

Ethernet ist wegen des Layer 2 – Verfahrens nicht deterministisch.

1.4 Folgende Frage ist in einem Simatic-Forum aufgetaucht :

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

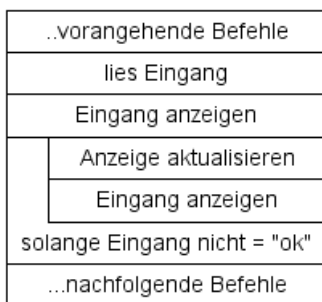
Was antworten Sie ?

Die Hardware ist in Ordnung, Ethernet kann als Layer 1/2-Basis benutzt werden.

Profinet I/ ist aber ein reines Datenkommunikationsprotokoll.

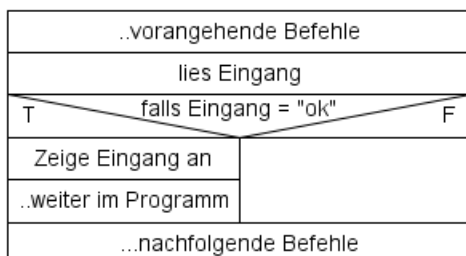
Zur Kommunikation des Simatic Managers mit einer SPS dient das S7-Protokoll.

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



In der Polling-Schleife muß der Wert der geprüften Variablen „Eingang“ gelesen werden !

b) Was halten Sie von dieser Verbesserung (Begründung !) :



Hier fehlt die Polling-Schleife.

Falls „Eingang“ nicht den gewünschten Wert hat, wird nicht erneut gefragt !

1.6 Welche Aussagen sind richtig ?

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

1.7 Welche Aussagen sind richtig ?

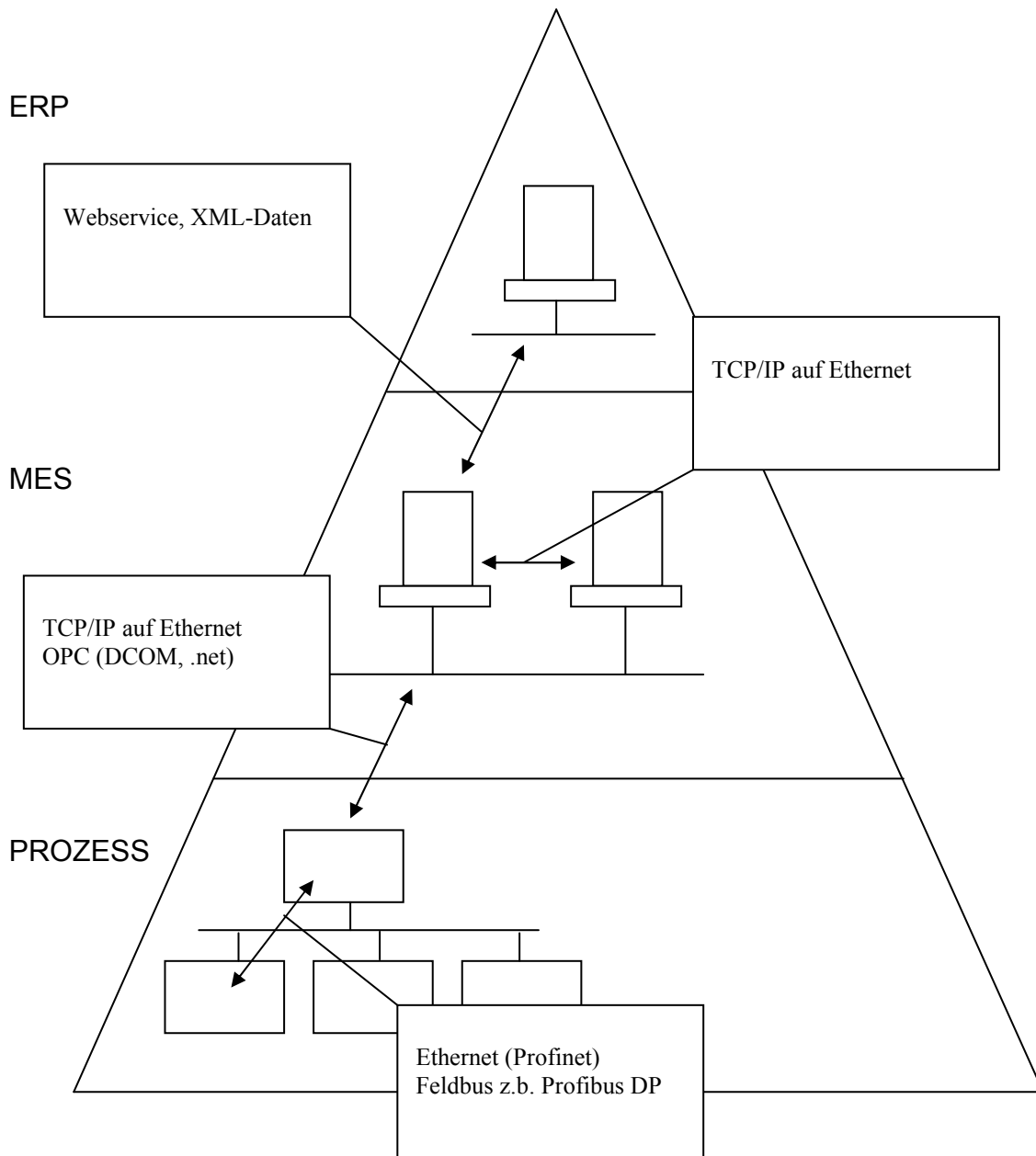
- In Profibus DP ist keine direkte Master-Slave Kommunikation möglich
- Profibus DP ist nur mit Master-SPS funktionsfähig
- In Ethernet mit TCP/IP muß stets ein Gerät als Master betrieben werden
- Profibus benötigt zur Signalübertragung nur eine zweipolige Twisted-Pair Leitung

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 ?

Polling-Schleifen sind im PC nötig, weil ein Programm vom Betriebssystem nur einmal ausgeführt wird und dann terminiert. Will man öfter nach einem Wert fragen, muß dies im Anwenderprogramm geschehen.

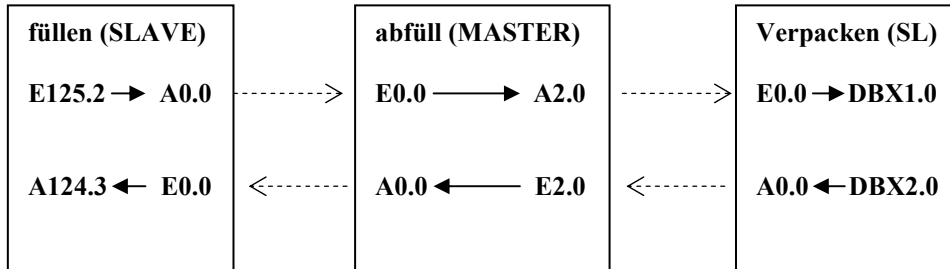
Die SPS liefert diese Wiederholung durch das Betriebssystem, das das Anwenderprogramm automatisch in einem Zyklus wiederholt.

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



# Teil 2, mit Unterlagen

1.



**füllen : U E125.2**  
 = A0.0  
 U E0.0  
 = A124.3

**abfüll : U E0.0**  
 = A2.0  
 U E2.0  
 = A0.0

**verpacken : U E0.0**  
 = DB10.DBX1.0  
 U DB10.DBX2.0  
 = A0.0

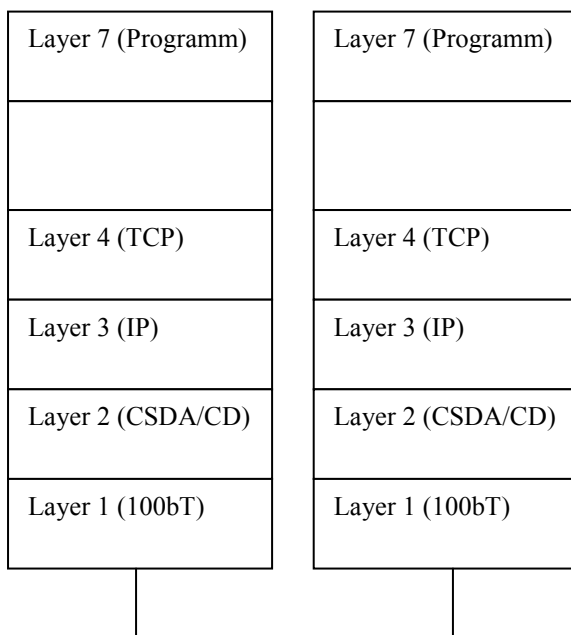
2.

Ja !

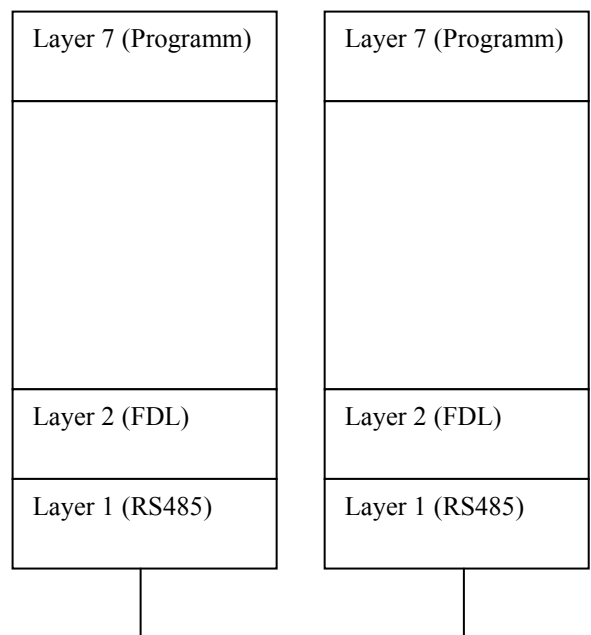
Die Maximaldauer ist die Summe der drei Zykluszeiten der SPS plus die Tokenlaufzeit im Profibus.

3.

a)



b)



4.

In IRT wird durch eine synchrone Uhr in allen Teilnehmern (PTP : precision time protocol) im Ethernetverfahren ein Zeitschlitz eingebaut, in dem direkte Kommunikation Point-to-Point stattfindet. Das ist dann keine Buskommunikation, hat auch nichts mit Ethernet zu tun.

5.

In Profinet I/O wird ein Provider/Consumer Modell eingesetzt, das dem Master/Slave aus DP entspricht.

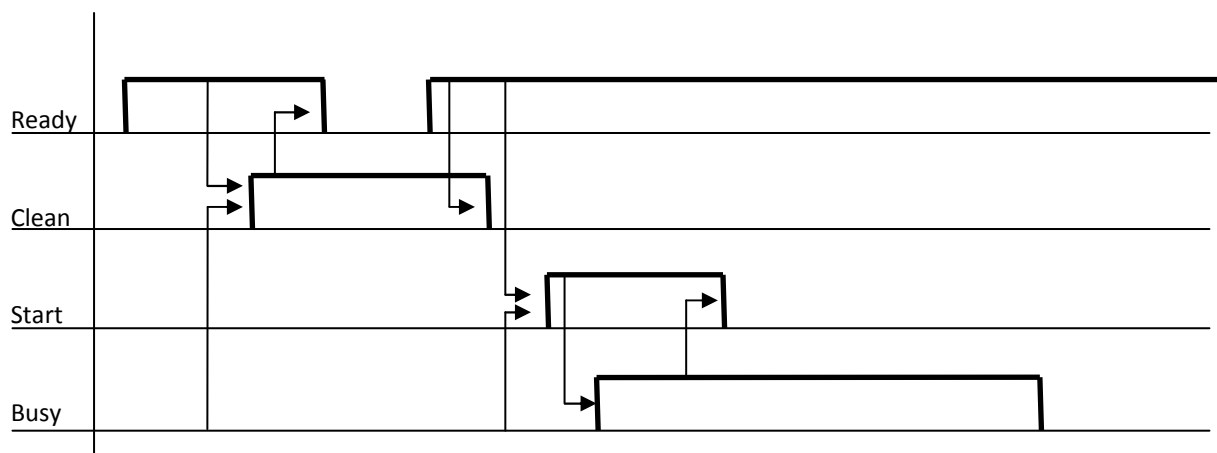
a) Das ist technisch nicht nötig, Ethernet braucht keinen Master

b) Es soll Investitionsschutz bringen, d.h. die alten Softwarestrukturen aus DP bleiben erhalten

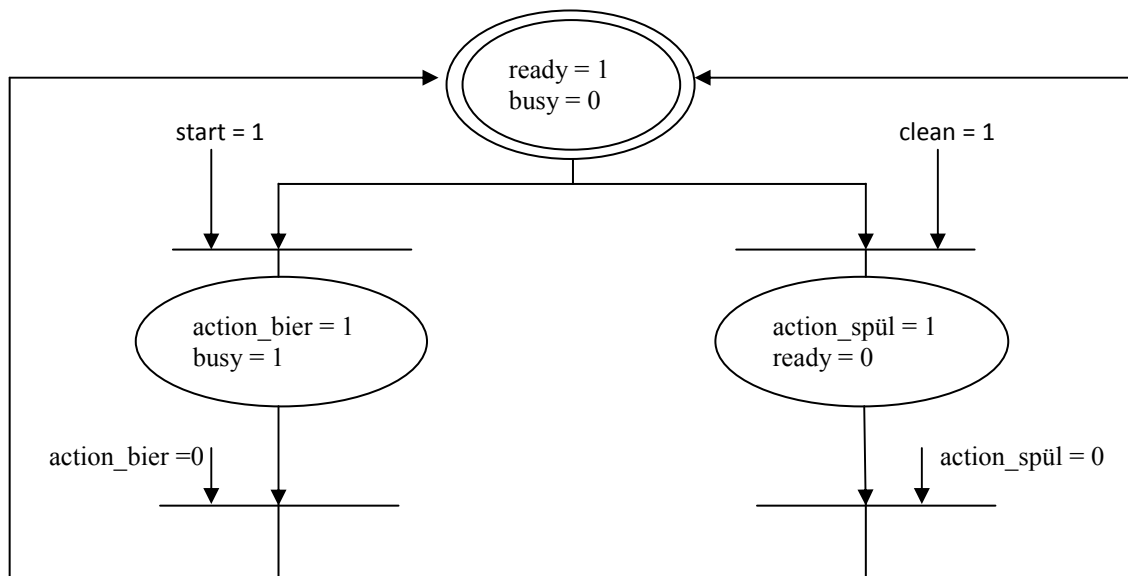
6.

In Profinet CBA wird der DCOM-Kommunikationskern aus den alten Microsoft-Betriebssystemen (bis XP) in die SPS implementiert. Distributed Component Object Model ermöglicht, ohne umständliche Konfiguration auf TCP/IP zu kommunizieren, auch mit PC's (-> OPC).

7.



8.



9.

lies READY										
T	falls READY = 1	F								
lies BUSY										
T	falls BUSY = 0	F								
<table border="1"> <tr><td>CLEAN = 1</td></tr> <tr><td>lies READY</td></tr> <tr><td>solange READY = 1</td></tr> <tr><td>lies READY</td></tr> <tr><td>textbox1.text = "running"</td></tr> <tr><td>solange READY = 0</td></tr> <tr><td>CLEAN = 0</td></tr> <tr><td>textbox1.text = "done"</td></tr> </table>	CLEAN = 1	lies READY	solange READY = 1	lies READY	textbox1.text = "running"	solange READY = 0	CLEAN = 0	textbox1.text = "done"	textbox1.text = "SPS arbeitet"	textbox1.text = "SPS nicht bereit"
CLEAN = 1										
lies READY										
solange READY = 1										
lies READY										
textbox1.text = "running"										
solange READY = 0										
CLEAN = 0										
textbox1.text = "done"										



10.

referenziere "bier" als objekt
referenziere "zutat" als objekt
referenziere "abfüll" als objekt
bier.ip zuweisen
zutat.ip zuweisen
abfüll.ip zuweisen
starte bier.spülen
lies bier.status
solange spült
biermenge = 150
zutat = lemon
zutatmenge = 12
starte bier
starte zutat
lies bier.status
lies zutat.status
solange bierstatus = running or zutatstatus = running
starte abfüll

**Imports bier.dll**

**Public class testprogramm**

**Private bier as new universalmodul**

**Private zutat as new universalmodul**

**Private abfüll as new universalmodul**

**Bier.ip = "62.245.200.163"**

**Zutat.ip = "62.245.200.164"**

**Abfüll.ip = "62.245.200.165"**

**Private sub button\_click() handles click**

**Bier.start\_clean()**

**Do**

**Bier.read\_status()**

**Loop until bier.status = "ready"**

**Bier.quantity = 150**

**Zutat.product = "lemon"**

**Zutat.quantity = 12**

**Bier.start\_product()**

**Zutat.start\_product()**

**Do**

**Bier.read\_status()**

**Zutat.read\_status()**

**Loop until bier.status= "ready" AND zutat.status = "ready"**

**Abfüll.start\_product()**

**End sub**

**End class**

12.

```
<? xml version = "1.0" encoding = "UTF-8" ?>  
<anfrage>  
  <auftrag> ja </auftrag>  
  <biermenge> 150 </biermenge>  
  <zutat> lemon </zutat>  
  <zutatmenge> 12 </zutatmenge>  
  <etikett> meier20.jpg </etikett>  
</anfrage>
```