

Fachschule für Elektrotechnik, Maschinenbautechnik und Metallbautechnik
der Landeshauptstadt München



Klasse :

Name :

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 ?

.....

.....

.....

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

.....

.....

.....

.....

.....

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 ?

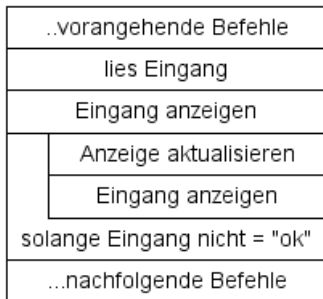
.....

.....

.....

.....

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



.....

.....

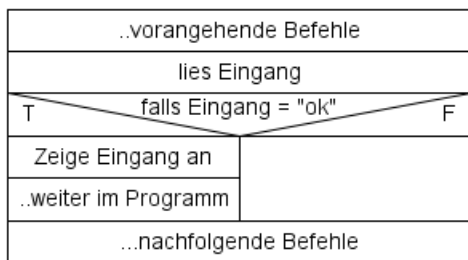
.....

.....

.....

.....

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



.....

.....

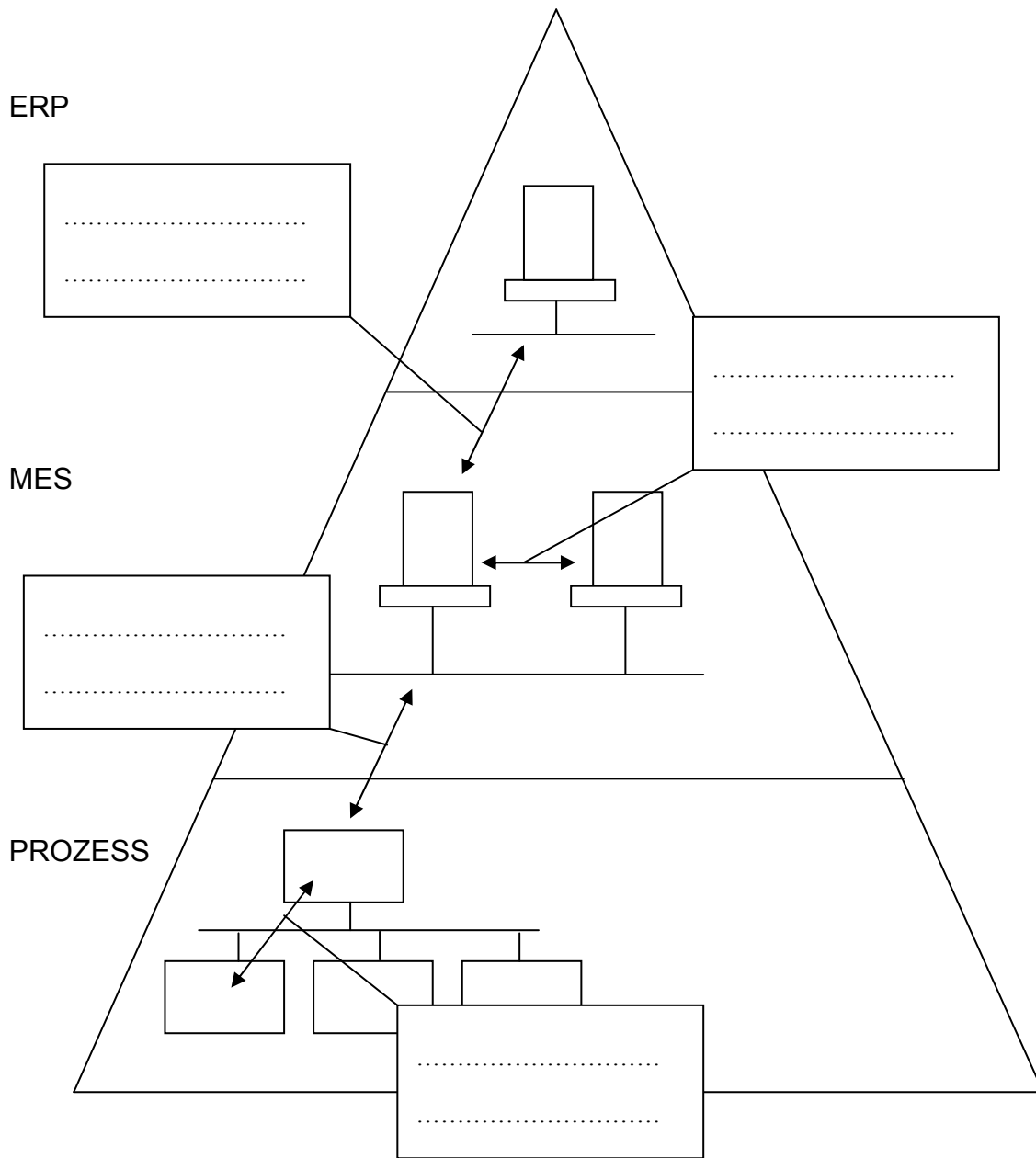
.....

.....

.....

.....

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

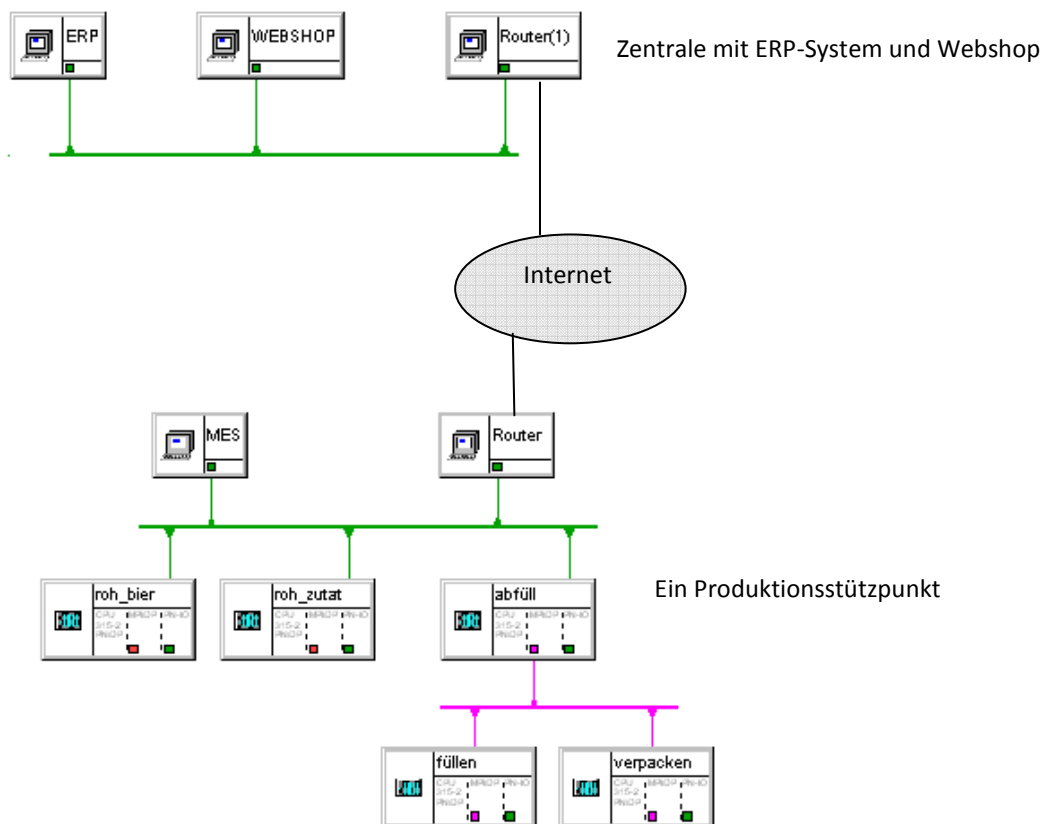


Anlagendokumentation , gesamte Anlagenstruktur :

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 qualifiziertes 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 :

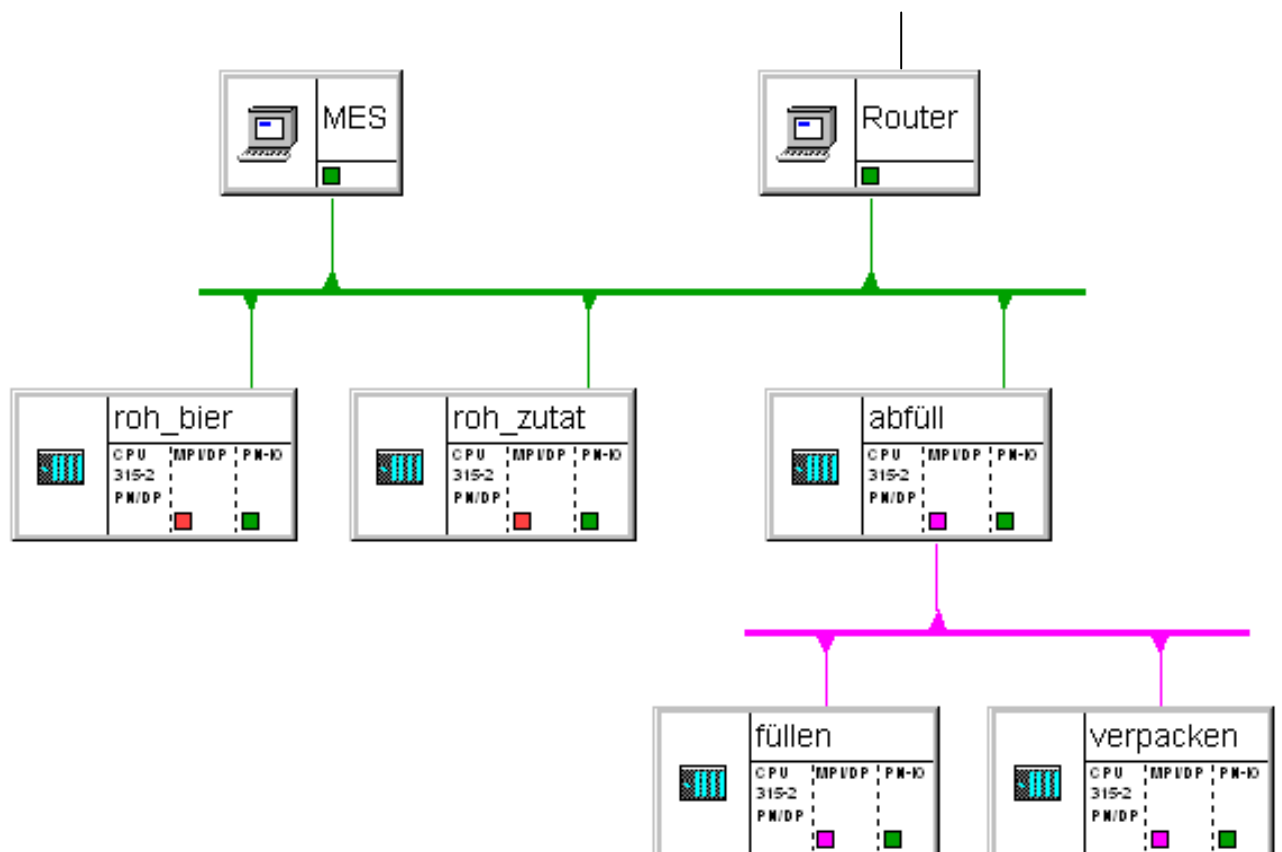


Anlagendokumentation, Aufbau der Produktionsstützpunkte:

An den Produktionsstätten ist überall die gleiche Anlage aufgebaut :

Ein MES-System unter Windows2008 steuert über TCP/IP drei angeschlossene 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 :

Eigenschaften - DP Slave

Allgemein | Kopplung | Konfiguration

Zeile	Mode	Partner-DP-Adr	Partner-Adr	lokale Adr	Länge	Konsistenz
1	MS	1	A 0	E 0	1 Wort	Einheit
2	MS	1	E 0	A 0	1 Wort	Einheit

Anschluß der Slave-SPS verpacken am Master abfüll :

Eigenschaften - DP Slave

Allgemein | Kopplung | Konfiguration

Zeile	Mode	Partner-DP-Adr	Partner-Adr	lokale Adr	Länge	Konsistenz
1	MS	1	A 2	E 0	1 Wort	Einheit
2	MS	1	E 2	A 0	1 Wort	Einheit

Teil 2, mit Unterlagen

Prozessebene :

- Die SPS **füllen** steuert den Produktfluß in die Abfüllstation für die Getränkedosen.
- Ein Sensor an **E125.2** dieser SPS meldet, wenn die aktuelle Dose gefüllt ist.
- Dieses Signal wird direkt an die SPS **verpacken** übertragen und dort in **DB10.DBX1.0** abgelegt.
- Die SPS schließt dann die Dose, leitet sie zur Etikettierung weiter und stellt die nächste Dose bereit. Dies wird dann durch das Bit **DB10.DBX2.0** an die SPS **füllen** gemeldet, wo es direkt an Ausgang A124.3 gelangt.

Aufgaben :

1. Geben Sie alle Programme in den beteiligten SPS an, die nötig sind um die Übertragung von E125.2 von SPS **füllen** zur SPS **verpacken** und umgekehrt das Bit DB10.DBX2.0 aus **verpacken** an die SPS **füllen** zu übertragen.

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

2. Ist diese Übertragung deterministisch ?
Wenn ja : aus welchen Bestandteilen kann die Maximaldauer der „Reaktionszeit“ ermittelt werden ?

Für die nächste Teilaufgabe wird nun angenommen, daß :

- a) **roh_bier** und **roh_zutat** direkt über OpenTCP 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.

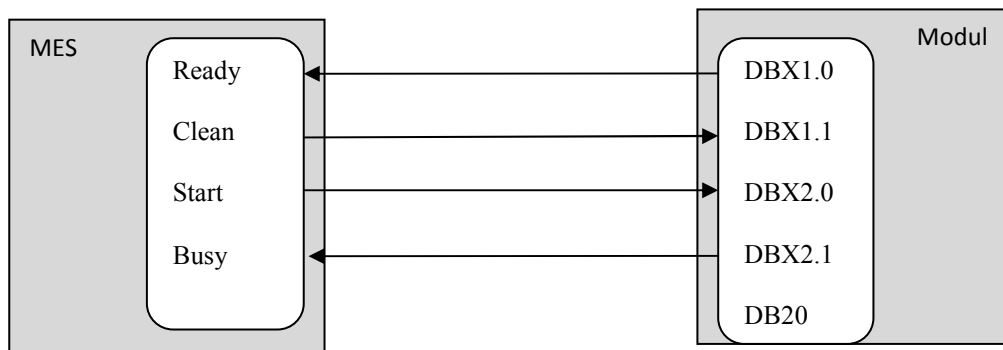
6. Jetzt läuft als Kommunikationskern DCOM. Erklären sie kurz, was hinter dieser Bezeichnung steckt (keine reine Übersetzung der Abkürzung bitte !)

MES-Ebene :

Die beiden Module zur Bereitung der Rohstoffe werden von MES mit einem einfachen Start/Busy-Handshake gesteuert.

MES setzt START, das Modul antwortet mit BUSY, welches sowohl als Quittung für START als auch als Funktionsdaueranzeige (bleibt 1, solange Modul arbeitet) dient.

Weiter kann MES ein Signal CLEAN an ein Module (oder beide) absetzen. Die Anzeige der Betriebsbereitschaft des Moduls READY geht damit auf 0, und das Modul beginnt, alle Rohrleitungen rückzuspülen. Wenn der Vorgang erfolgreich abgeschlossen werden konnte, geht READY wieder auf 1, was MES veranlasst, auch CLEAN wieder zurückzusetzen.



7. Zeichnen Sie ein Timing-Diagramm, das alle Signale zwischen MES und einem Modul darstellt. Mit Pfeilen tragen Sie bitte die Signalbeziehungen ein.

In der SPS **roh_bier** werden die beiden Vorgänge „bierherstellen“ und „rückspülen“ durch zwei action-Signale gesteuert.

action_bier startet ein Programmmodul, das Bier herstellt, dieses meldet dann durch **/action_bier** (wird rückgesetzt), daß der Vorgang abgeschlossen ist.

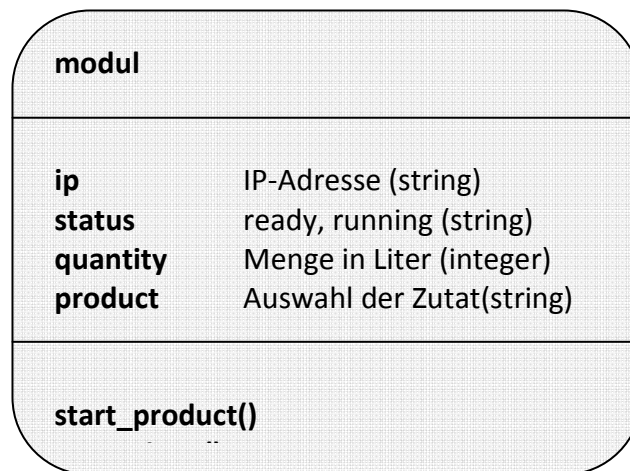
action_spül startet ein Programmmodul, das rückspült, dieses meldet dann durch **/action_spül**, daß der Vorgang abgeschlossen ist.

8. Geben Sie mit einem Petrinetz oder einem Graph für S7 den Programmteil in der SPS **roh_bier** an, der den Handshake mit MES abwickelt und über die beiden action-Signale die Programmabläufe zur Produktionssteuerung startet.

In MES wird die Behandlung der Peripherieschnittstellen später objektorientiert abgewickelt. Zunächst (oder zur Entwicklung der Klassenbibliothek) soll aber der Handshake auch auf dieser Seite als Standardprogramm betrachtet werden :

9. Geben Sie den Struktogrammteil an, der, ausgelöst durch einen Bedienerbutton, den Handshake zur Steuerung des Rückspülvorgangs aus MES realisiert.

Im Weiteren steht zur Ansteuerung der SPS-Module in MES eine Klassenbibliothek **beer.dll** zur Verfügung. Diese beinhaltet u.a. die Klasse **modul** :



ip ist die IP-Adresse der SPS im QDN-Format (z.b. : "196.246.245.34")

status meldet den Betriebszustand der SPS :

ready : Modul ist funktionsbereit

running: Modul produziert gerade oder spült

product gibt an, welche Zutat ggf. in den Mischtank gefüllt werden soll

quantity gibt an, welche Menge der Zutat ggf. in den Mischtank gefüllt werden soll

start_product() ist die Methode zur Beauftragung eines Produktionslaufs

start_clean() ist die Methode zur Beauftragung eines Rückspülvorgangs

read_status() ist die Methode zum Auslesen des Betriebsstatus

10. Geben Sie ein Struktogramm an, das zunächst einen Spülvorgang durchführt , dann 150 liter Bier und parallel 12 liter der Zutat "lemon" in den Mischtank befüllt, und dann einen Abfüllvorgang auslöst.
Die Vorgänge laufen asynchron, das bedeutet, daß ihr Ende durch Abfragen des Status ermittelt werden muß.

11. Codieren Sie das Programm aus Aufgabe 10 in VB .net.
Der Vorgang soll durch einen Button ausgelöst werden.
Geben Sie den Programmkopf mit allen Zuweisungen 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 .jpg-files (z.b. „meier20.jpg“)

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