

Technikerschule - Fachschule für  
Maschinenbau-, Metallbau-, Informatik- und Elektrotechnik  
der Landeshauptstadt München



Technikerprüfung 2015/16

Automatisierungstechnik

Zeit : 150 Minuten

Klasse :

Name :

	Punkte:	Note :	Unterschriften:
Erstkorrektur			
Zweitkorrektur			

## Teil 1 , ohne Unterlagen

**Klasse :**

**Name :**

### 1. Kreuzen Sie hier alle richtigen Aussagen an :

- MAP hat als erstes modernes Netz CSMA/CD in Layer 2
- Im ISO/OSI-Modell wird in Layer 4 die Transportsprache (z.b. XML) definiert
- Der SAP identifiziert einzelne Dienste in ISO/OSI- Layern
  
- Verfügbarkeit ist ein Maß für Funktionswahrscheinlichkeit
- Vorbeugende Wartung verlängert die Phase der Zufallsausfälle
- Burn-In verlagert die Phase der Frühausfälle vom Kunden zum Hersteller
- Dynamische Redundanz ist teurer, aber sicherer als statische
  
- In Profibus DP wird durch eindeutige Adressvergabe Routing möglich
- Im Layer 3 von Profibus DP befindet sich das Routing-Protokoll
- Profibus DP ist schneller als Ethernet
- In Profinet IRT ist deterministische Kommunikation möglich
- Profinet I/O und Profinet IRT sind in Layer 2 identisch
  
- Webservices können XML oder Binärdaten liefern
- OPCuA basiert auf DCOM und ist deshalb Windows7-kompatibel
- OPC ist ein Middleware-Protokoll
- URL ist die Abkürzung für Unified Remote Language
- HTML und XML sind ähnliche Sprachen
  
- MES-Systeme speichern zentral die wichtigsten Daten einer Firma
- MES und ERP kommunizieren oft mit Webservices
- ERP ist wegen erwünschter Redundanz nie der einzige Speicherort für Stücklisten

Beantworten Sie die folgenden Fragen auf dem karierten Lösungsbogen.

Pro Antwort sollten 3 oder 4 Zeilen genügen :

1. Was verstehen Sie unter dem Begriff Determinismus ?
2. Was bedeutet der Begriff : "just-in-sequence" ?
3. Was bedeutet der Begriff "vertikale Kommunikation" ?
4. Wieso wird bei Anlagen mit Industrie 4.0-Struktur vertikale Kommunikation benötigt ?
5. Beantworten Sie folgende Frage aus einem SPS-Forum :  
  
„Sind TCP/IP und Ethernet identische Protokolle ?“
6. In einem Handshakeprotokoll werden als Signalhandshake von einem PC ein START-Signal zu einer SPS geschickt, zurück ein ACKNOWLEDGE-Signal. (Ablauf einmalig, also nicht zyklisch). Geben Sie in beliebigen, aber geeigneten Sprachen (auch grafisch) den jeweils nötigen Programmteil in PC und SPS an.
7. Worin besteht der Unterschied zwischen einer Variablen und einem Tag in OPCuA ?

8. Beschreiben Sie ,wie der deterministische Kommunikationsteil in Profinet IRT funktioniert.

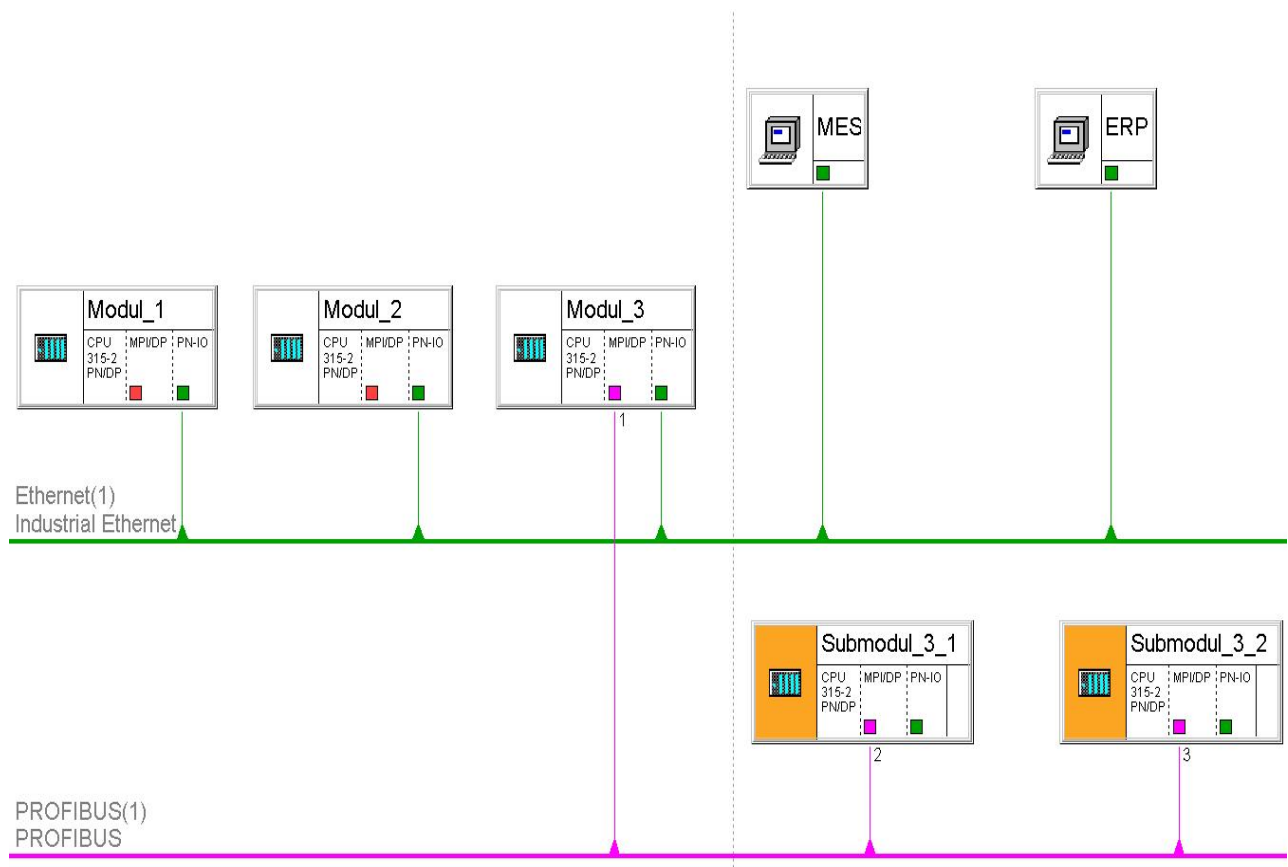
9. Was unterscheidet ein Netz von einem Bus ?

10. Zeichnen sie in 2 Timing-Skizzen einen Signalhandshake und einen Funktionshandshake. Zeigen Sie durch Pfeile die Abhängigkeiten der Signale. Erklären sie mit einem kurzen Text den Unterschied zwischen den beiden Handshakevarianten. Gehen Sie dabei auch auf den Begriff "relevantes" Funktionsende" ein.

## Teil 2, mit Unterlagen

Alle nachfolgenden Aufgaben bearbeiten Sie bitte auf dem karierten Lösungsblatt !

Wir betrachten die hier abgebildete Anlage :



Die abgebildete Anlage kann in Losgröße 1 auftragsgeführt Produkte herstellen.

Dies geschieht in drei Fertigungsmodulen (Modul\_1 bis Modul\_3).

In Modul\_3 werden 2 Hilfsprozesse durch jeweils eine eigene SPS gesteuert, die mit Profibus DP an die Master-SPS Modul\_3 angekoppelt sind.

Ein MES-System verkettet die Fertigungsmodule, es holt sich seine Produktaufträge aus der SQL-Datenbank des ERP-Servers.

### **1. Aufgabe :**

Alle Ethernetteilnehmer kommunizieren mittels TCP/IP.

Die Geräte sind folgendermaßen, zum Teil fehlerhaft, konfiguriert :

Modul\_3 : 212.17.210.1/20  
Modul\_2 : 212.17.209.1/20  
Modul\_1 : 212.17.206.20/20  
MES : 212.17.208.255/20  
ERP : 212.17.223.254/20

**Prüfen Sie, welche Geräte im Firmennetz 212.17.208.0 mit der Netmask 255.255.240.0 nicht funktionieren können, und konfigurieren sie diese auf funktionsfähige IP-Adressen.**

### **2. Aufgabe :**

Die Kommunikation zwischen Prozessebene und MES-System wird mit einem Middlewareprotokoll im Intranet realisiert. Es kommt OPCuA zum Einsatz, der auf einem passenden Server installiert wird, und zwar die Binärprotokollvariante. Der OPC-Server "mes1" läuft auf SAP 3800.

**Geben Sie die URL an, mit der in einem graphischen OPC-Client dieser OPC-Server angesprochen werden kann.**

Die Kommunikation zwischen der Master-SPS Modul\_3 und den Slaves Submodul\_3\_1 und Submodul\_3\_2 ist als Handshakeprotokoll auszuführen. Hierbei wird ein einfacher Start/Acknowledge-Handshake realisiert.

#### Protokollbeschreibung :

Modul\_3 startet das jeweilige Submodul mit einem **START**- Signal.

Ein Submodul antwortet mit **ACKNOWLEDGE** als (Signal-)Handshakesignal und setzt **READY=0**

Ein Submodul meldet **READY=1**, wenn seine mechanische Funktion beendet ist, Modul\_3 kann es dann wieder starten (womit Ready dann wieder Null wird).

### **3. Aufgabe :**

Modul\_3 soll nun die beiden Submodule parallel starr koppeln (zyklischer Ablauf).

**Geben Sie in einer beliebigen aber hier geeigneten Programmiersprache (auch grafisch) die Programmfunktion in Modul\_3 und einem Submodul an.**

An Modul\_3 befindet sich ein Taster, der einem Bediener ermöglicht, an den beiden Submodulen ein optisches Signal (Warnlampe) zu betätigen.

Der Taster ist an Modul\_3 am Eingang 124.0 kontaktiert, an den Submodulen sind die Signallampen an A 125.3 angeschlossen.

Die Konfiguration der Kommunikationszellen in den SPS ist den folgenden Bildern zu entnehmen.

Modul\_3 -> Submodul\_3\_1 :

Zeile	Mode	Partner-DP-Adr	Partner-Adr	lokale Adr	Länge	Konsistenz
1	MS	1	A 1	E 10	1 Wort	Einheit
2	MS	1	E 1	A 10	1 Wort	Einheit

Modul\_3 -> Submodul\_3\_2 :

Zeile	Mode	Partner-DP-Adr	Partner-Adr	lokale Adr	Länge	Konsistenz
1	MS	1	A 3	E 10	1 Wort	Einheit
2	MS	1	E 3	A 20	1 Wort	Einheit

#### **4. Aufgabe :**

**Geben Sie in AWL in allen 3 SPS das Programmstück an, das für die beschriebene Funktion nötig ist.**

Zwischen Modul\_3 und MES wird ein komplexer Handshake ausgeführt. Ein Bediener startet den Ablauf, der nur einmalig durchlaufen wird, mit einem Button-Druck.

Beschreibung :

MES prüft, ob die SPS READY (=1) meldet. Ist das nicht der Fall, wird nicht gewartet, sondern die Funktion sofort abgebrochen.

Wenn READY = 1, erfolgt von MES ein START-Signal (=1) mit einer Sonderfunktion :

Mit erkanntem START setzt die SPS READY auf 0  
Damit geht START wieder auf 0  
Damit setzt die SPS READY wieder 1  
START geht in Folge auf 1

Nur wenn dieser Ablauf korrekt durchgeführt wurde, erfolgt zwischen START und ACKNOWLEDGE ein Funktionshandshake nach üblichem Muster.

(dieser Ablauf stellt sicher, daß z.B. Ready nicht fehlerhaft 1 liegt, obwohl die SPS gar nicht funktioniert, beide Pegel (0 und 1) werden aktiv getestet)

Die Kommunikation wird direkt mit OPC-Variablen durchgeführt.  
Die OPC-Variablen in MES heißen so wie die in den SPS, also **START**, **ACKNOWLEDGE** und **READY**.

Die Anfangsbedingung durch den Bediener-Button muß in Aufgabe 5 und 6 nicht berücksichtigt werden, er startet den Ablauf ohne in der Logik vorzukommen.

5. Aufgabe :

Zeichnen Sie ein Timingdiagramm für diesen Handshake.

6. Aufgabe :

Geben Sie in einer beliebigen aber hier geeigneten Programmiersprache (auch grafisch) in MES das Programmstück an, das für die Ansteuerung vom Modul\_3 (Handshake) nötig ist.



Im Weiteren wird die Kommunikation mit einer Klassenbibliothek TP\_2015.dll erleichtert, die hierfür folgende Klassen bereitstellt :

<b>Name :</b>	ERP_CONNECT
<b>Attribute :</b>	ID (Integer) Modul (String) Variante (Byte) Exists (Bool)
<b>Methoden :</b>	Lies_teil()

- ID** : Unter der ID werden die Produkte gespeichert (ohne Lücken fortlaufend).
- Modul** : gibt an, für welches Modul Info gelesen werden soll (Modul\_1 bis Modul\_3)
- Variante** : gibt an, welche Fertigungsvariante zu bestücken ist (1-3 möglich).
- Exists** : gibt an, ob ein Produkt dieser ID in der Datenbank vorhanden ist.

**Lies\_teil()** : ermittelt die Information für Exists und Variante. Hierzu ist vorher ID und Modul zu mit Werten zu besetzen.

<b>Name :</b>	MODUL_CONTROL
<b>Attribute :</b>	Modul_IP (String) Variante (Byte) Status (String)
<b>Methoden :</b>	Lies_Status() Start_Modul()

- Modul\_IP** : Module müssen mit ihren IP-Adressen (xxx . xxx . xxx . xxx) parametrier werden.
- Variante** : Hier wird dem Modul mitgeteilt, welche Fertigungsvariante durchzuführen ist.
- Status** : Betriebszustand des Moduls. Möglich sind hier : RUNNING und OK

**Start\_Modul()** : führt den Start-Handshake mit der SPS aus.

**Lies\_Status()** : liest aus der SPS den aktuellen Betriebszustand

**WICHTIG** : Es wird jetzt nicht mehr direkt mit den OPC-Variablen gearbeitet, sondern nur noch mit den Werkzeugen dieser Klassenbibliothek !

Um diese Prüfung zu erleichtern, wird nun ein Testbetrieb angenommen :

Der Anlagenfahrer gibt am MES-System an einer Textbox1 eine ID ein, ab der eine Charge von 12 Produkten gefertigt werden soll. Dann gibt er den Vorgang durch Drücken eines Buttons frei.

Die MES-Software startet die Programmausführung mit diesem Buttonclick.

Der Fertigungsvorgang wird aber erst angefahren, sobald in der ERP-Datenbank eine Charge von mindestens 12 Produkten ab der eingegeben ID vorhanden ist. Sonst wird gewartet.

Für den "Prüfungsbetrieb" werden dann aber nur die Bestückungsaktionen durch Modul 3 ausgeführt. (Die Produkte sind dergestalt, daß eine Bearbeitung nur durch Modul\_3 möglich ist). Nach der Charge von 12 Teilen steht die Produktion wieder.

**7. Aufgabe :**

**Geben Sie in einer beliebigen Programmiersprache (Code, kein Struktogramm !) in MES das Programmstück an, das für die beschriebene Funktion nötig ist.**

**8. Aufgabe :**

**Kennzeichnen Sie durch beschriftete Pfeile an je einem Beispiel, welche Syntaxelemente in ihrem Programmcode ein OBJEKT, ein ATTRIBUT, eine KLASSE und eine METHODE darstellen.**