

Fachhochschule München Fachbereich Elektrotechnik
Übungsaufgaben aus Echtzeitsysteme

Aufgabe 1: (Zuverlässigkeit und Sicherheit) (20P.):

Eine Prozeßrechenanlage besteht aus den in folgender Tabelle aufgelisteten Systemkomponenten mit der jeweiligen (Dauer-)Unverfügbarkeit q :

Komponente:	Unverfügbarkeit q:	Komponente:	Unverfügbarkeit q:
CPU	0.01	RAM	0.01
DMA-Interface	0.02	Analog-E/A Baugruppe	0.03
Terminal	0.02	Drucker	0.005

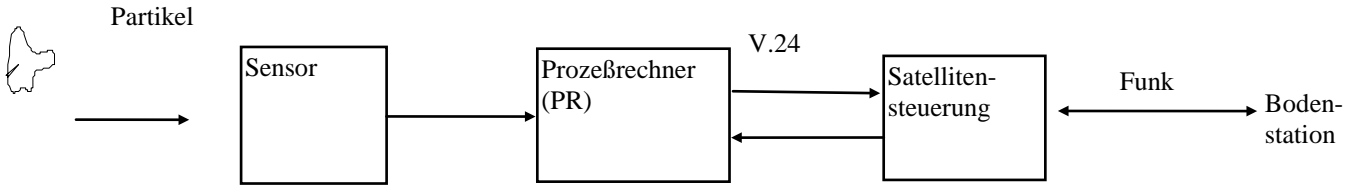
Aufgabe 1.1: Die Anlage sei verfügbar, wenn die CPU, das RAM, das DMA-Interface, die Analog-E/A Baugruppe, der Drucker und das Terminal funktioniert. Zeichnen Sie das Zuverlässigkeitsersatzschaltbild und berechnen sie die Unverfügbarkeit q_{ges} des Gesamtsystems!(6P.)

Aufgabe 1.2: Um eine höhere Verfügbarkeit zu erhalten, sollen einige (oder eine einzelne) Komponenten der Anlage als Doppelsubsystem ausgeführt. (Für die dazu notwendige Koppel- und Vergleichselektronik wird eine Verfügbarkeit von 1 angenommen und eine automatische Fehlererkennbarkeit sei vorausgesetzt!) Welche Komponenten zu verdoppeln bringt dabei den größten Beitrag zur Erhöhung der Verfügbarkeit? Welche maximale verbesserte Verfügbarkeit läßt sich erreichen, wenn zwei Komponenten (welche?) verdoppelt werden können? (6P.)

Aufgabe 1.3: Um die Anlage bei einer Störung in einen **sicheren** Zustand zu bringen, seien die CPU und das DMA-Interface notwendig. Welche Strategie der Komponentenverdoppelung ist nun am sinnvollsten und unter welchen Umständen muß die Anlage in den sicheren Zustand gebracht werden, um eine Verbesserung der Sicherheit gegenüber 1.2 zu erreichen? Zeichnen Sie das neue Ersatzschaltbild ! Wie groß ist jetzt die Verfügbarkeit der Anlage? (8P.)

2. Aufgabe: Satellitenexperiment - Staubzähler - (25 P.)

Im Weltraum sollen permanent Staubpartikel mittels eines gegebenen Sensors in einem Satelliten gezählt werden. Der in dieser Aufgabe betrachtete **Prozeßrechner (PR)** des Staubzähl-Experiments **im Satelliten** hat die Aufgabe, die durch den Sensor gewonnenen **Meßdaten zu erfassen, sie auszuwerten** und alle **Ergebnisse auf Anfrage** an die **Bodenstation** auf der Erde **zu schicken**. Zu Veranschaulichung dient das folgende Systemschaubild:



Die **Bodenstation** ist der Master der **den PR über** entsprechende **Ein-Byte-Kommandos fernsteuert**. Diese Kommandos werden im Satelliten zunächst durch die Satellitensteuerung entgegengenommen und mittels serieller Verbindung (V.24) an den **PR** weitergereicht. Umgekehrt sendet der **PR** eine entsprechende Antwort an die Satellitensteuerung, die diese dann an die Bodenstation funkt. Diese serielle Verbindung wird mit der Geschwindigkeit von 1000 Bytes/s betrieben. Der eingesetzte **PR** ist mit Realzeitbetriebssystem OS9000 mit Prioritätsscheduling ausgerüstet. Das Senden und der Empfang eines Bytes kann gleichzeitig geschehen. Das Abholen des empfangenen Bytes und das Bereitstellen des zu sendenden Bytes kostet je 1ms Verarbeitungszeit. Der Empfang eines Bytes wird durch das Setzen des Events **BYTE_IN =1** signalisiert, und nach Abholen des Bytes automatisch rückgesetzt. Es darf das nächste Byte zum Senden übergeben werden, wenn das vorherige gesendete Byte vollständig übertragen wurde; was durch das Bit **BYTE_OUT=1** angezeigt wird. Es wird zunächst vorgeschlagen die auszuführenden Aufgaben auf die folgenden Tasks zu verteilen:

- **Task A:** . Bei bestehendem Funkkontakt der Satellitensteuerung mit der Erde wird diese Task A alle 10 Sekunden von Task C aktiviert ansonsten ruht sie: Die Task A ist **Sende- und Empfangstask in Einem**. Sie wartet auf ein Kommando, interpretiert es **und** führt es aus. Nach Erhalt und Interpretation des Kommandos schickt sie sofort eine Quittungsnachricht (Länge 1 Byte). Bei Kommando **MESSW** folgen dann noch alle vorliegenden Meßdaten (zwischen 1 - 100 Meßwerte je 16 Byte). Bei Kommando **RESET** muß sich der Prozeßrechner **sofort** (innerhalb von 20 ms) zurücksetzen. Die Interpretation eines beliebigen Kommandos Task A dauert maximal 1 ms. Nach 6 Sekunden beendet sich die Task A in jedem Fall.
- **Task B:** Der intelligente Meßwertaufnehmer (Sensor) liefert alle 5 Minuten einen Satz von Meßwerten. Diese Meßwerte werden zunächst von Task B entgegengenommen und ausgewertet. Es entstehen so alle fünf Minuten zwischen 1 Meßwert (16 Byte) und 100 Meßwerten (1600 Byte). Die Ergebnisse dieser Auswertung werden dann in einem reservierten Speicherbereich (**shared-memory**) abgelegt. Von dort werden sie von Task A abgeholt. Die Task B benötigt in allen Fällen eine maximale Rechenzeit von 1 Sekunde und darf jederzeit unterbrochen werden.
- **Task C:** Die Task C dient als Timertask und muß außerdem zyklisch wiederkehrend System-Selbsttestaufgaben durchführen. Task C wird bei Bedarf jede ms geweckt und ist aktiv. Task C erhöht alle 100ms zunächst einen Zeitzähler; dazu benötigt sie 1ms Rechenzeit. Nach Ablauf jeder Sekunde muß Task C innerhalb 100ms zusätzlich Diagnoseaufgaben: Test auf Funkkontakt zur Erde, System-Selbsttest und Watchdogreset durchführen; dazu benötigt sie 20ms Rechenzeit. Bei Funkkontakt zur Erde startet sie alle 10 Sekunden die Task A; dafür benötigt sie 10ms Rechenzeit.

Aufgabe 2.1: Wie groß ist jeweils die Auslastung je verursacht durch Task A,B und C (Rechenweg muß ersichtlich sein!!) (5 P.)

A= _____ B= _____ C= _____

Für Aufgabe 2.2: Es sei Funkkontakt und es fällt zunächst alle 5 Minuten nur 1 Meßwert an.

Aufgabe 2.2: Teilen Sie für diesen Fall den drei Tasks A, B und C Prioritäten (0=hohe,1=mittlere,2=niedrige Priorität) zu und begründen Sie kurz Ihre Entscheidung (2 P.)

	Priorität	Begründung
TASK A		
TASK B		
TASK C		

Für restliche Aufgaben: Es sei Funkkontakt und es fallen alle 5 Minuten 1-100 Meßwerte an.

Aufgabe 2.3: Zeigen Sie an Hand der Zustandsereignisdiagramme von Task A,B und C das Zusammenspiel der Tasks zur Lösung der Aufgaben. (8 P)

Aufgabe 2.4: Wenn 100 Messwerte angefordert sind, kann dann jede Task bei gegebener Aufgaben- und Ihrer gewählten Prioritätsverteilung rechtzeitig seine Aufgaben erledigen? (Begründung) (5 P.)

Aufgabe 2.5: Verteilen die Aufgaben und Prioritäten zwischen Task A und C so, dass die Echtzeitbedingungen eingehalten werden. (5 P.)
