

AN025**Application Note 025 zu MODULAR-4 (ML3 und ML5)****Messen der CPU-Auslastung im Echtzeitbetrieb**

Autor: HB

AN025.DOC (6 Seiten)

1. Definition

Im folgenden wie auch in allen anderen Sorcus Schriften werden folgende Kurzbezeichnungen verwendet:

Bezeichnung	Karte	Bussystem	PC
ML1	Multi-LAB	ISA/EISA	IBM-PC oder AT
ML3	MODULAR-4/Z280	Micro-Channel	PS/2
ML4	MODULAR-4/Z80	ISA/EISA	IBM-PC oder AT
ML5	MODULAR-4/Z280	ISA/EISA	IBM-PC oder AT
ML6	MODULAR-4/386	16-Bit ISA/EISA	IBM-PC/AT
ML8	MODULAR-4/486	16-Bit ISA/EISA	IBM-PC/AT

2. Aufgabenstellung

Bei Multi-Tasking Echtzeitbetriebssystemen stellt sich häufig die Frage, ob die Leistungsfähigkeit eines Systems unter allen Einsatz-Bedingungen ausreicht. Eine Abschätzung ist bei der Planung eines Systems hilfreich, eine exakte Messung und möglichst auch eine kontinuierliche Überwachung des Systems wären aber in jedem Fall wünschenswert.

3. Lösung

Das Multi-Tasking Betriebssystem auf den MODULAR-4 Karten erlaubt das kontinuierliche Messen und Überwachen der aktuellen Auslastung der CPU unter allen Einsatzbedingungen. Das Prinzip ist dabei folgendes:

Zur Erinnerung:

Es gibt zwei Task-Typen: Interrupt-Tasks (I-Tasks) und Nicht-Interrupt-Tasks (NI-Tasks). Die Unterscheidung der verschiedenen Interrupt-Tasks ist hier ohne Bedeutung. Eine Interrupt-Task wird aufgerufen, wenn die entsprechende Interrupt-Bedingung auftritt. Die Nicht-Interrupt-Tasks werden vom Task-Scheduler des Betriebssystems reihum aufgerufen, sofern sie aktiviert sind. Eine mehrfach aktivierte NI-Task wird auch mehrfach aufgerufen. NI-Tasks laufen im Hintergrund und sind von allen Interrupt-Tasks unterbrechbar.

Es wird nun gemessen, wie oft pro Zeiteinheit der Task-Scheduler eine einmal aktivierte NI-Task aufrufen konnte.

Die Zahl der Aufrufe sinkt naturgemäß bei vielen Interrupts oder bei vielen anderen aktiven NI-Tasks. Sie sinkt ebenfalls bei intensiver Kommunikation zwischen Host-Rechner und Karte.

Wenn keine NI-Task installiert ist, wird die Zahl der Leerdurchläufe des Task-Schedulers gemessen.

Auch bestimmte Programmierfehler können entdeckt werden: Normalerweise müssen NI-Tasks die Kontrolle aktiv an das Betriebssystem zurückgeben. Dies wird aber manchmal vergessen. Das führt dazu, daß andere NI-Tasks nicht mehr aufgerufen werden. Als Ergebnis der Messung würde in diesem Fall 0 angezeigt.

Für die Messung wird auf der Karte das Systemprogramm 253 (FDH) unter Task 15 installiert. Es führt die Messung in festen Zeitintervallen durch (1/64 sec oder 1 sec). Der aktuell gemessene Wert wird mit dem seit Start des Programms gemessenen Minimum und Maximum verglichen. Wenn der neue Wert größer als das aktuelle Maximum ist, wird er als neues Maximum gespeichert, wenn er kleiner ist als das aktuelle Minimum, wird er als neues Minimum gespeichert.

Die seit Start des Programms 253 aufgetretenen Spitzenwerte können zu jeder Zeit vom PC bzw. Host-Rechner ausgelesen werden.

Die zusätzliche Belastung durch das Meßverfahren beträgt nur wenige Mikrosekunden pro Meßzeitintervall, so daß die Messung auch kontinuierlich eingeschaltet bleiben kann. Das Programm 253 kann aber auch durch Aufruf der Prozedur 8 ganz abgebrochen werden.

Umrechnung der Meßwerte in Prozent CPU-Auslastung

Der aktuelle Meßwert sowie Minimum und Maximum können auch in Prozent CPU-Auslastung umgerechnet werden. Dadurch erhält man aussagefähigere Werte. Allerdings ist hierbei zu beachten, daß das Multi-Tasking-Betriebssystem auf der Karte ein, was wir so nennen, "kooperatives Multi-Tasking" macht. Das heißt, daß die CPU den einzelnen Tasks nicht feste Zeiteinheiten zuteilt (Zeitscheiben), sondern daß die Tasks die Dauer, für die sie die CPU benutzen, selbst bestimmen. Wenn die CPU wenig ausgelastet ist, wird die Zeit genutzt, um die NI-Tasks häufiger aufzurufen. Wenn die CPU mehr ausgelastet ist, werden die NI-Tasks entsprechend seltener aufgerufen. Wenn eine NI-Task nicht mehr so oft aufgerufen wird, wie von der Anwendung gefordert, dann ist die CPU überlastet.

Allerdings ergibt sich diese Auslastungsgrenze (= Mindestanzahl erforderlicher Aufrufe der NI-Tasks) nur aus der jeweiligen Anwendung. Wenn diese Mindestanzahl bekannt ist, kann auch die prozentuale Auslastung ermittelt werden.

Eine Ausnahme gibt es hiervon: Wenn keine NI-Task installiert ist und nur I-Tasks, also II- oder DI-Tasks, laufen, dann ist die Mindestanzahl = 1.

Ein Beispiel soll die Ermittlung der Mindestanzahl verdeutlichen: Wenn z.B. über eine serielle Schnittstelle Zeichen empfangen werden, so wird das üblicherweise mit einer Interrupt-Task gemacht, z.B. werden die Zeichen in einen Puffer geschrieben. Anschließend werden die Zeichen im Puffer dann von einer NI-Task analysiert und es wird entsprechend reagiert. Es ist klar, daß die NI-Task pro Zeichen mindestens einmal aufgerufen werden muß, um das Zeichen zu verarbeiten. Bei einer Baudrate von 9600 kann maximal ca. jede Millisekunde ein Zeichen ankommen. Also sind bei einer Meßdauer von 1 Sekunde mindesten 1000 Durchläufe der NI-Task erforderlich, damit unter ungünstigsten Bedingungen alle Zeichen bearbeitet werden. Zeigt dieses Programm nun bei einer Meßdauer von 1 Sekunde z.B. 8000 an, so beträgt die aktuelle CPU-Auslastung = $1000 / 8000 * 100 \% = 12,5\%$.

Wenn nun noch weitere serielle Schnittstellen bedient werden sollen, die mit der gleichen oder einer niedrigeren Baudrate arbeiten, so ändert das nichts an der geforderten Mindestanzahl von 1000. Arbeitet eine Schnittstelle mit höherer Baudrate, so muß sie für die Kalkulation der Mindestanzahl herangezogen werden.

Allgemein errechnet sich die prozentuale Auslastung:

$$\text{CPU-Auslastung (in \%)} = \frac{\text{Mindestanzahl Durchläufe} * 100}{\text{aktueller Meßwert Durchläufe}}$$

Es können sich dabei auch Werte über 100% ergeben. Dann ist die CPU überlastet. Außer dem aktuellen Meßwert können so auch Minimum und Maximum der CPU-Auslastung in Prozent ermittelt werden.

Zu beachten ist dabei, daß die Berechnung der prozentualen CPU-Auslastung unabhängig von der Meßdauer ist. Allerdings kann es sein, daß bei großer Meßdauer kurzzeitige Spitzenbelastungen oder Überlastungen der CPU nicht erkannt werden. Die geeignete Meßdauer muß sich deshalb nach der jeweiligen Anwendung und nach der Anforderung an die Messung richten.

Eine Besonderheit ist dadurch gegeben, daß eine NI-Task mehrmals aktiviert werden kann. Sie wird dann bei einem Durchlauf auch mehrmals aufgerufen. Das ist bei der Abschätzung der Mindestanzahl zu berücksichtigen. Wenn bei dem oben genannten Beispiel die NI-Task z.B. zweimal aktiviert wird, kann als Mindestanzahl 500 angegeben werden.

Hinweis

Die CPU-Auslastung ist naturgemäß auch abhängig von der Verwendung des Cache. Mit Hilfe des Makro-Befehls 28H kann das Cache Control Register der CPU eingestellt werden, mit dem Makro-Befehl 29H kann es zurückgelesen werden:

Cache abschalten:	28H FDH 12H 60H 00H
Cache für Instruktionen:	28H FDH 12H 20H 00H
Cache für Daten:	28H FDH 12H 40H 00H
Cache für Instruktionen und Daten:	28H FDH 12H 00H 00H
Cache Control Register (CCR) lesen:	29H FDH 12H
Anwort (x ist ungültig):	29H CCR x

Leider kann auch hier keine für alle Anwendungen optimale Einstellung angegeben werden. Der Cache sollte aber eingeschaltet werden. Welches die günstigste Einstellung ist, muß im Einzelfall ermittelt werden.

4. Das Programm 253

Das Programm 253, Vers. 2.A., ist ab Betriebssystem-Version ML5-3RU.004 bzw. ML3-3RU.004 im EPROM enthalten. Es wird unter Task 15 installiert, es benötigt keinen Datenbereich.

Der Start des Programms erfolgt durch Aufruf von Prozedur 6. Nach maximal zwei Meßintervallen steht ein Ergebnis zur Verfügung. Der Status (Parameter 0) ist dann = 3. Alle Ergebnisse müssen als Parameterworte gelesen werden. Außer dem Ergebnis der letzten Messung stehen auch Minimum und Maximum zur Verfügung.

4.1. Die Prozeduren

Prozedur	Funktion
6	Start des Programms. Die Parameter 2 bis 7 werden auf die Anfangswerte gesetzt. Diese Prozedur kann auch verwendet werden, um das Programm neu zu starten.
8	Abbrechen des Programms. Die letzten Ergebnisse bleiben gespeichert.

4.2. Die Parameter

Alle 2-Byte Parameter müssen als Parameter-Worte gelesen und geschrieben werden.

Parameter	Init	Bedeutung
0	0	Status: 0 = Programm installiert, nicht aktiv 1 = Programm mit Prozedur 6 gestartet 2 = Erste Messung läuft, Ergebnisse noch nicht gültig 3 = Kontinuierliche Messung läuft, Ergebnisse sind gültig 4 = Programm mit Prozedur 8 abgebrochen 5 = Fehler aufgetreten
1 <1>	4	Meßintervall: 0 = 1/64 Sekunde (= 15,625 ms) 4 = 1 Sekunde 8 = 1 Minute 12 = 1 Stunde
2, 3	0	Aktueller Meßwert: Meßwert = 0: Die CPU ist zu 100% ausgelastet Meßwert > 65280: Sehr geringe CPU-Auslastung
4, 5	0	Bisher gemessenes Maximum
6, 7	65534	Bisher gemessenes Minimum
8, 9 <2>	0	Schwelle für Minimum. Wenn diese Schwelle unterschritten wird, wird Parameter 10 ausgewertet.
10 <2>	0	Reaktionen bei Alarm: 0 = keine Reaktion 1 = Kontroll-LED einschalten 2 = Service-Request-Byte zum PC senden <3> (löst PC-Interrupt aus) 3 = Kontroll-LED und Service-Request <3>
11 <2>	0	Service-Request-Byte <3>

<1> Eine Änderung dieses Parameters wird erst nach Neustart des Programms, z.B. durch Aufruf der Prozedur 6, wirksam.

<2> Eine Änderung dieses Parameters wird sofort wirksam.

<3> Bei Version 2.A. noch nicht implementiert