Application Note AN052

Watchdog und Spannungsüberwachung auf der MODULAR-4/486 Karte

Autor: JD Datei: AN052.DOC (4 Seiten)

Watchdog

In komplexen Anwendungen der MODULAR-4/486, bei denen eine Vielzahl von Echtzeitprogrammen unterschiedlichste Aufgaben zu erledigen haben, muß häufig sichergestellt sein, daß alle Tasks in einer bestimmten Zykluszeit abgearbeitet werden. Als ein Hilfsmittel für die Überwachung des pünktlichen Aufrufs von Tasks stellt die MODULAR-4/486 Basiskarte einen Watchdog-Timer zur Verfügung. Dieser muß durch eine Task in regelmäßigen Abständen neu gesetzt werden (Re-Triggern). Um ein gewünschtes Zeitschema einzuhalten, sollte die Nach-Triggerung an der "zeitkritischsten" Stelle aller installierten Tasks erfolgen (nur an einer Stelle!).

In Anwendungen, in denen die Interrupt-Belastung der MODULAR-4 sehr hoch ist, kann es z.B. dazu kommen, daß NI-Tasks nur sehr selten vom Betriebssystem aufgerufen werden. Soll überwacht werden, daß eine bestimmte NI-Task trotzdem rechtzeitig an die Reihe kommt, muß die Re-Triggerung des Watchdog in der Haupt-Prozedur dieser NI-Task stattfinden.

Kommt das Betriebssystem nicht dazu, die Task, die die Re-Triggerung übernimmt, rechtzeitig aufzurufen, weil ein Programm abgestürzt ist, oder andere Tasks zu viel Zeit beanspruchen, bleibt die Re-Triggerung aus. Dadurch läuft der Watchdog-Timer ab. Die Basiskarte reagiert darauf je nach Revision unterschiedlich:

- **Rev. B**: Auf der MODULAR-4 Karte wird ein lokaler Reset ausgelöst (*lokal* bedeutet hier, der Reset ist auf die Basiskarte und aufgesteckte Module beschränkt und hat keine Auswirkungen auf den Host-PC). Alle Programme werden abgebrochen.
- **Rev. C:** Auf der MODULAR-4 Karte wird ein NMI-Interrupt ausgelöst. Wenn für diesen Interrupt eine Interrupt-Task installiert ist und der NMI freigegeben ist, ruft das OsX die Haupt-Prozedur dieser Task auf. Dort besteht die Möglichkeit, einen sicheren Betriebszustand herzustellen, Daten zu retten oder den Host-PC durch einen Service-Request zu informieren. Durch einen Lesezugriff auf die lokale I/O-Adresse 28h kann die Ursache des NMI-Interrupts herausgefunden werden (s.u.). Der Watchdog kann danach durch Re-Triggerung wieder neu aktiviert werden.
  Soll auch eine Rev.C Karte mit einem lokalen Reset auf ein Ablaufen des Watchdog reagieren, besteht dafür die Möglichkeit, am Stecker St1 die Pins 6 (Reset-Eingang) und 13 (Watchdog-Ausgang) zu verbinden.

In zukünftigen Versionen des Betriebssystems werden erweiterte Diagnosemöglichkeiten zur Tasküberwachung enthalten sein, z.B. Betriebssystem-Parameter, die die Nummer der zuletzt aufgerufenen Task enthalten.

Um die Funktionalität des Watchdog nutzen zu können, sind folgende Schritte erforderlich:

1. Das Jumperfeld J4 der Basiskarte muß entsprechend dem Handbuch (Kapitel 2.2.4) gesteckt werden, um den Watchdog zu aktivieren.
2. Die Laufzeit des Watchdog-Timers kann ebenfalls mit dem Jumperfeld J4 zwischen den Werten 1,6 s und 100 ms ausgewählt werden. Dabei ist zu beachten, daß die Jumper-Stellung bei Rev. C und Rev. B Karten genau umgekehrt ist (siehe Handbuch). Auf Wunsch kann werksseitig auch eine andere Zeit eingestellt werden.
3. Wenn diese Einstellungen gemacht sind, ist der Watchdog aktiv. Das bewirkt im allgemeinen zunächst jedoch noch gar nichts, da in der Default-Einstellung das Betriebssystem die Nach-Triggerung des Watchdog übernimmt. Um dieses abzuschalten, ist der Betriebssystem-Parameter 214 (Byte) =0 zu setzen (Auto-Retrigger-Off).
4. Anstelle des Betriebssystems muß jetzt eine Anwender-Task die Nach-Triggerung des Watchdog übernehmen. Das kann mit Hilfe der Bibliotheksfunktion **ml8rt_trigger_watchdog** erledigt werden. Zu beachten ist, daß jeder Zugriff auf die Echtzeituhr (RTC) den Watchdog Nach-triggert.
5. Bei Rev.C Karten sollte eine DI- oder II-Task unter dem NMI-Interrupt (02) installiert werden, um auf einen Ablauf des Watchdog-Timers reagieren zu können. Zu beachten ist, daß der NMI-Interrupt demaskiert werden muß (**ml8rt_unmask_int(NMI)**).

Überwachung von TI-Tasks

Die Haupt-Prozedur einer TI-Task wird vom TI-Scheduler in dem bei ihrer Aktivierung (**wakeup_ti_task**) angegebenen Zeitraster aufgerufen. Um zu überwachen, ob dies ordnungsgemäß geschieht, stellt das Betriebssystem über den für alle Task-Arten gültigen Watchdog hinaus speziell für TI-Tasks weitere Möglichkeiten zur Verfügung. Die Dauer eines Timer-Ticks wird im Betriebssytem-Parameter 316 (Doppelwort) in µs (Default-Wert =0, entspricht 1 ms) angegeben.

Wenn der TI-Task-Scheduler eine TI-Task nicht zur geplanten Zeit aufrufen kann, weil eine andere Task noch aktiv ist, so wird im Betriebssystem-Parameter 334 (Wort) jeweils der Zeitverzug (in Timer-Ticks) abgespeichert, mit dem die Task an die Reihe kommt. Im Parameter 320 wird die seit dem letzten Reset der Karte aufgetretene maximale Verzugszeit (in Timer-Ticks) beim Aufruf von TI-Tasks gehalten. Die Nummer der TI-Task, die für diese Verzögerung verantwortlich ist, wird in Parameter 322 (Wort) eingetragen.

Wenn der Zeitverzug einen einstellbaren Grenzwert überschreitet (Parameter 324 (Wort): Zeit in Timer-Ticks, FFFFh = keine Meldung), kann dies mit einem Service-Request signalisiert werden. Das in einem solchen Fall zu sendende SRQ-Wort steht im Betriebssystem-Parameter 326 (Wort) und kann vom Anwender definiert werden.

Welche TI-Task gerade an der Reihe ist, kann dem Parameter 336 (Wort) entnommen werden.

Alle hier aufgeführten Parameter können durch Anwender-Tasks ausgelesen werden. Auf diese Weise kann jederzeit geprüft werden, ob TI-Tasks planmäßig abgearbeitet werden. Außerdem kann bei Überschreiten eines Grenzwertes für die Verzögerung automatisch mit einem Service-Request reagiert werden.

Spannungsüberwachung

Die MODULAR-4/486 Basiskarte stellt differenzierte Möglichkeiten zur Überwachung der Versorgungsspannung zur Verfügung.

Die Versorgungsspannung kann in zwei Stufen überwacht werden:

1. Auf ein Unterschreiten der Schwellspannung 4,8V (Rev.C) kann mit einem NMI reagiert werden. Damit dies möglich ist, müssen auf dem Jumperfeld J4 die Pins 3 und 4 verbunden werden (siehe Handbuch, Kapitel 2.2.4). Bei MODULAR-4/486 Rev. B ist die Schaltschwelle für das Auslösen eines NMI mit Hilfe eines Potentiometers zwischen 4,4V und 5,8V einstellbar (Vorgehen: s. Handbuch, Kapitel 3.2.2.4).
   Rev. B Karten benutzen für die Spannungsüberwachung Pin 1 von Stecker St1 als Eingang. Damit ist es möglich, eine externe Spannung zu überwachen, die an Pin 1 angeschlossen wird. Um die +5V Versorgungsspannung der Karte zu überwachen, ist Pin 1 mit dem daneben liegenden Pin 3 (+5 V) zu verbinden.
   In einer unter dem NMI installierten Interrupt-Task kann versucht werden, wichtige Daten zu retten oder einen sicheren Betriebszustand herzustellen, bevor die Spannung soweit abgesunken ist, daß ein lokaler Reset ausgelöst wird (s. 2). Durch einen Lesezugriff auf die lokale I/O-Adresse 28h kann die Ursache des NMI herausgefunden werden (s.u.).
2. Sinkt die Versorgungsspannung unter die Schwelle von ca. 4,65V (Rev.B: 4,4V), löst die Spannungsüberwachung einen lokalen Hardware-Reset aus.

Auslesen der NMI-Interruptquelle

Werden mehrere NMI-Quellen freigegeben, kann die Ursache eines NMI durch einen 16-Bit-Lesezugriff auf die lokale I/O-Adresse 28h herausgefunden werden. Darin ist:

- Bit 9 des gelesenen Statuswortes =0 (!), wenn die Spannungsüberwachung einen NMI ausgelöst hat.
- Bit 10 =1, wenn der Watchdog einen NMI verursacht hat (nur bei Rev. C).

Die NMI-Leitung geht darüber hinaus über den SP-Bus an alle aufgesteckten SPB-Module. Einige dieser Module erlauben die Anwahl des NMI für ihre Zwecke. Ob ein Modul der Verursacher eines NMI war, läßt sich von den jeweiligen Modulen auslesen. Wie dies geschieht, ist bei den einzelnen Modulen unterschiedlich und in den zugehörigen Handbüchern zu finden.

Reset-Verhalten

Ein durch den Watchdog oder die Spannungsüberwachung verursachter lokaler Reset bewirkt den Abbruch aller Programme. Im Gegensatz zu einem von der PC-Bibliothek ausgelösten Reset (**ml8_reset**) werden daraufhin jedoch keine Initialisierungen entsprechend den EEPROM-Eintragungen vorgenommen. Im Anschluß an den Reset ist nur das Mini-OS auf der Karte aktiv, welches nicht in der Lage ist, die Initialisierungen durchzuführen.

Die Reset-Ursache kann ausgelesen werden.

Ein Ausschnitt aus zwei C-Beispielprogrammen soll den Gebrauch des Watchdog aufzeigen:

```
/*************************************************************************/
// auto_init-Prozedur einer Task, in der der Watchdog aktiviert wird und der
// NMI-Interrupt für eine Watchdog-Reaktion freigegeben wird.
/*************************************************************************/

void auto_init(void)
{
   ml8rt_entry();                      // save registers and prepare
   ml8rt_write_par_byte(0,214,0);      // disable auto-retrigger
   ml8rt_trigger_watchdog();           // trigger watchdog
   ml8rt_unmask_int(NMI);              // enable NMI
   ml8rt_exit();                       // restore registers
}

/*************************************************************************/
// Haupt-Prozedur einer NI-Task, die den Watchdog nach-triggert.
/*************************************************************************/

void main_task()
{
   ml8rt_entry();                      // save registers and prepare
   ml8rt_trigger_watchdog();           // trigger watchdog
   ml8rt_exit();                       // restore registers
}



:::::::::::::::::::::::::::::::: neue Task: ::::::::::::::::::::::::::::::::



/*************************************************************************/
// Haupt-Prozedur einer DI-Task, die unter dem NMI als Reaktion auf einen
// Ablauf des Watchdog zu installieren ist.
/*************************************************************************/

void main_task(void)
{
   ml8rt_entry();                      // save registers and prepare
   // ... user defined reaction to watchdog timeout (re-init, srq, etc.)
   ml8rt_trigger_watchdog();           // reset and restart watchdog
   ml8rt_exit_interrupt();             // restore registers
}
```