Ingenieurbüro / Vertrieb
Klaus Kohl
Postfach 1173 - 86406 Mering
Zeppelinstr. 10 - 86415 Mering
Tel. 08233/30524
Fax 08233/9971

PC-Meßtechnik

Kennung: PC-MT06
Datum: 17.10.96
Stichworte: PC, Meßtechnik, AT-Bus, Meßkarten

Klaus Kohl

PC-Meßkarten

Im vorerst letzten Teil der PC-Meßtechnikserie schließt sich der Kreis um die an einem PC verfügbaren Schnittstellen und wir kehren zu dem zurück, was dem Rechner erst seine vielfältigen Anwendungen ermöglicht. Bei der Beschreibung der IBM-kompatiblen PC/AT-Slots und dessen Verwendung für Meßkarten sollen auch die Interrupt- und DMA-Leitungen und dessen Ansteuerung nicht zu kurz kommen.

8Bit-Slot

GND	B1	A1	-I/O CHCK
RESET DRV	B2	A2	D7
+5V	B3	A3	D6
IRQ2	B4	A4	D5
-5V	B5	A5	D4
DRQ2	B6	A6	D3
-12V	B7	A7	D2
Reserviert	B8	A8	D1
+12V	B9	A9	D0
GND	B10	A10	I/O CH RDY
-MEMW	B11	A11	AEN
-MEMR	B12	A12	A19
-IOW	B13	A13	A18
-IOR	B14	A14	A17
-DACK3	B15	A15	A16
DRQ3	B16	A16	A15
-DACK1	B17	A17	A14
DRQ1	B18	A18	A13
-DACK0	B19	A19	A12
CLOCK	B20	A20	A11
IRQ7	B21	A21	A10
IRQ6	B22	A22	A9
IRQ5	B23	A23	A8
IRQ4	B24	A24	A7
IRQ3	B25	A25	A6
-DACK2	B26	A26	A5
T/C	B27	A27	A4
ALE	B28	A28	A3
+5V	B29	A29	A2
OSC	B30	A30	A1
GND	B31	A31	A0

16Bit-Slot

MEM CS16	D1	C1	SBHE
-I/O CS16	D2	C2	LA23
IRQ10	D3	C3	LA22
IRQ11	D4	C4	LA21
IRQ12	D5	C5	LA20
IRQ15	D6	C6	LA19
IRQ14	D7	C7	LA18
-DACK0	D8	C8	LA17
DRQ0	D9	C9	-MEMR
-DACK5	D10	C10	-MEMW
DRQ5	D11	C11	SD08
-DACK6	D12	C12	SD09
DRQ6	D13	C13	SD10
-DACK7	D14	C14	SD11
DRQ7	D15	C15	SD12
+5V	D16	C16	SD13
-MASTER	D17	C17	SD14
GND	D18	C18	SD15

1. Der PC/AT-Slot

Da das Mainboard des ersten PC im August 1981 mit einem 8088-Prozessor, 64Kbyte RAM (Standardausstatung, erweiterbar auf 640K), 40Kbyte ROM und der notwendigen Verbindungslogik einschließlich Timer, DMA- und Tastaturcontroller schon sehr gefüllt war, wurde entschieden, die optionalen Erweiterungen wie Schnittstellen (seriell und parallel) und Grafikkarten nicht fest einzubauen, sondern als steckbare Ergänzungen vorzusehen. Um sich die Arbeit einfach zu machen, legte man einfach die notwendigen Prozessorsignale für Speicher- und I/O-Zugriff sowie die freien Interrupt- und DMA-Leitungen auf 5 sogenannten Slots, die jeweils eine Platine mit ebenen Kontaktflächen aufnehmen kann. Die erste, meist mitgelieferte Karte war eine Grafikkarte.

Obwohl inzwischen schon die unterschiedlichsten PC-Busse wie VESA Local Bus, Microchannel und PCI eingesetzt wurden und werden, bleiben die hier beschriebenen Slotkarten auch in der Meßtechnik die verbreitesten PC-Erweiterungen.

1.1. Pinbelegung

Da zuerst nur ein Prozessor mit 8Bit-Datenbreite verwendet wurde, genügte auch ein 8Bit-Erweiterungsslot mit 2*31-poligen Klammern. Erst nach Einführung des 80286-Prozessors und den anderen Erweiterungen des PC-AT wie zusätzliche Daten- und Adreßleitungen, weiteren Interrupts und den für 16Bit-Übertragungen gedachten DMA-Kanäle war eine Erweiterung nötig. Diese wurde glücklicherweise so gewählt, daß einfach eine zweite, 2*18polige Klammer den 8Bit-Slot fortsetzt.

1.2. Einsatzmöglichkeiten

Auf dem PC-Markt verfügbare Karten zeigen die Nutzung der PC/AT-Slots:

- Reine I/O-Karten:
  Die parallele und serielle Schnittstellenkarten nutzen nur den I/O-Speicherbereich. Dazu genügt es, wenn neben den Adreßleitungen A0 bis A9 und den Datenleitungen D0 bis D7 nur noch -IOR (I/O-Read), -IOW (I/O-Write) und ALE (Address latch enable) berücksichtigt wird.
  Oft wird dann noch die Interruptleitungen IRQ3/4 (für RS232) bzw. IRQ5 oder 7 (LPT) verwendet.
- Grafikkarten:
  Diese Karten verwenden außer dem I/O-Bereich auch noch den Speicher unterhalb der 1MByte-Grenze für die Betriebssystem-Erweiterung und dem Videospeicher.
- DMA-Karten:
  Festplattencontroller liefern Daten schneller, als früher die CPU sie speichern konnte. Deshalb entwickelte man einen Baustein, der die Daten direkt in das RAM bringt. Von diesem Baustein sind zwei freie Kanäle und zwei weitere Acknowledge-Leitungen am Slot verfügbar.

Eine später nicht mehr erläuterte Leitung ist I/O CHCK. Damit kann die Karte dem Prozessor mit einem 0 signalisieren, daß ein Speicherfehler aufgetreten ist.

Da der PC im Normalfall auch bei einem 16Bit-Zugriff auf I/O und RAM die Daten byteweise holt, muß eine Slotkarte rechtzeitig an -MEM CS16 bzw. -I/O CS16 seine 16Bit-Breite anzeigen.

1.3. Freie I/O- und RAM-Bereiche

Natürlich kann nicht jede Karte einen beliebigen Adreßbereich für sich in Anspruch nehmen, weil sonst die Funktion des gesamten Systems gestört werden könnte. Auf Grund der PC-Entwicklung sind auf dem Mainboard schon Bereiche reserviert und wegen entsprechender BIOS- und Windows(95)-Unterstützung sollte man andere Adressen nur für bestimmte Karten nutzen. Der Rest ist dann unser Schlachtfeld für Meßkarten.

RAM-Bereiche

$000000-$09FFFF	640Kbyte RAM
$0A0000-$0BFFF	Speicher der Grafikkarte
$0C0000-$0C3FFF	BIOS der Grafikkarte
$0C4000-$0C7FFF	BIOS des Festplattencontrollers
$0C8000-$0EFFFF	Weitere BIOS oder Upper Memory
$0F0000-$0FFFFF	BIOS des Mainboards
$100000-$10FFEF	Upper Memory für DOS
$10FFF0-	weiteres RAM

Da der Speicher für Programme (hauptsächlich Spiele) unterhalb der 640K-Grenze sehr kostbar ist, wird der nicht benötige Speicher oberhalb der Grafikkarte für ständig benötigte Programme wie Tastaturtreiber oder Betriebssystem genutzt. Durch einen Trick kann der Prozessor selbst im 16Bit-Modus noch den Speicherbereich über der magischen 1MByte-Grenze nutzen. Dazu wird einfach zu einem Seitenregister des Prozessors (ist für die absolute Adresse mit 16 zu multiplizieren) der Offset des gewünschten Zeigers addiert. Dadurch kann er noch die Speicheradresse $FFFF0 + $FFFF = $10FFEF erreichen, falls das sogenannte A20-Gate mitspielt. Dieses Gate wird über das Betriebssystem verwaltet.

I/O-Bereiche

$000-$00F	DMA-Controller 1 (8237)
$010-$01F	reserviert (DMA im PS/2)
$020-$03F	Interruptcontroller 1 (8259)
$040-$05F	Timer (8253)
$060-$06F	Tastaturcontroller
$070-$07F	RTC und NMI-Maske
$080-$09F	DMA-Seitenregister
$0A0-$0BF	Interruptcontroller 2 (8259)
$0C0-$0DF	DMA-Controller 2 (8237)
$0E0-$0EF	reserviert für Systemplatine

$0F0-$0FF	Ansteuerung des Coprozessor 80287
$100-$1EF	**frei für Erweiterungen**
$1F0-$1FF	Festplattencontroller
$200-$20F	Gameport
$210-$21F	**frei für Erweiterungen**
$220-$22F	Soundblaster
$230-$26F	**frei für Erweiterungen**
$270-$27F	Parallelport 2
$280-$2AF	**frei für Erweiterungen**
$2B0-$2DF	EGA-Grafikkarte
$2E0-$2E7	frei für Erweiterungen
$2E8-$2EF	Serielle Schnittstelle 4
$2F0-$2FF	Serielle Schnittstelle 2
$300-$31F	Ethernetkarten (z.B. NE2000)
$320-$36F	**frei für Erweiterungen**
$370-$37F	Parallelport 1
$380-$3AF	Diverses
$3B0-$3BB	MDA-Grafikkarte (Herkules)
$3BC-$3BF	Parallelschnittstelle (z.B. im Compac)
$3C0-$3CF	EGA-Grafikkarte
$3D0-$3DF	CGA-Grafikkarte
$3E0-$3E7	**frei für Erweiterungen**
$3E8-$3EF	Serielle Schnittstelle 3
$3F0-$3F7	Floppy Controller
$3F8-$3FF	Serielle Schnittstelle 1

Hier muß man darauf achten, daß die meisten I/O-Karten nur die Adreßleitungen A0 bis A9 dekodieren und deshalb im restlichen Speicherbereich als Spiegelungen auftauchen. Auf der anderen Seiten dekodieren einige Bausteine wie DMA-Seitenregister oder Joystick die Adreßleitungen auch nicht vollständig und sind deshalb mehrfach im angegebenen Bereich vorhanden. Man muß sich vor Verwendung einer Karte immer vergewissern, ob der gewünschte Bereich leer ist.

1.4. Timing

Timing wird immer irgendwie mit einem Takt verbunden. In einem PC wird hier zwischen dem Prozessortakt und dem Bustakt unterschieden. Beginnend mit 4,77MHz am ersten PC stieg sowohl der Prozessortakt (jetzt schon bis 200MHz) als auch der Bustakt (standardmäßig ca. 8MHz für I/O) kontinuierlich. Für den Zugriff auf eine Karte ist aber die Zeit zwischen dem Anlegen aller relevanten Signale (z.B.: A0-A9, D0-D7, ALE und IOW) und der Übernahme der Daten von der I/O-Karte entscheidend. Diese Zeit kann meist im Setup des Mainboard eingestellt werden. Aber auch die Karte hat ein Mitspracherecht durch die Leitung I/O CH RDY (I/O Channel Ready). Solange sie auf 0 gezogen wird, wartet der Prozessor. Wird sie nicht angesteuert, so bleibt sie auf 1 und der Prozessor arbeitet mit den im Setup eingestellten Werten.

2. Interruptleitungen

Wir haben im Laufe dieser Serie schon mehrfach mit Interrupts zu tun gehabt. Neben dem auf der Grundplatine verwendeten Timer-Interrupt 0 haben wir auch schon die herausgeführten Interruptleitungen 3 und 4 bei der seriellen Schnittstelle genutzt. Im Uralt-PC gab es nur einen Interruptcontroller, der über 8 Interrupteingänge verfügt und an I/O-Adresse $20/$21 angesprochen wird. Im AT leistete man sich einen weiteren Controller mit weiteren 8 Eingängen. Weil man aber keine eigene Prozessorleitung spendieren wollte, werden diese Interrupts über Eingang IRQ2 durch den ersten Interruptcontroller geführt.

2.1. Verwendung im AT

Wie man an folgender Liste sieht, sind die auf den AT-Slot geführten Interrupts meist schon in festen Händen. Deshalb sollte für eigene Karte eher die Verwendung der selten genutzten IRQ10-IRQ12 und IRQ15 angestrebt werden.

Leitung	Interruptvektor	Verwendung
IRQ0	$08	Timer 0-Ausgang
IRQ1	$09	Tastatur
IRQ2	---	Verkettung mit Interrupt 8-15
IRQ3	$0B	Serielle Schnittstelle 2 / 4
IRQ4	$0C	Serielle Schnittstelle 1 / 3
IRQ5	$0D	Festplattenlaufwerk oder LPT2
IRQ6	$0E	Diskettenlaufwerk
IRQ7	$0F	LPT1
IRQ8	$70	Echtzeituhr (CMOS-RTC)
IRQ9	$0A/$71	Bildschirmsteuerung (alte IRQ2)
IRQ10	$72	frei (meist Ethernetkarten)
IRQ11	$73	frei
IRQ12	$74	frei (oder Mausschnittstelle)
IRQ13	$75	Coprozessor
IRQ14	$76	Festplattencontroller
IRQ15	$77	frei

Jedem Eingang ist ein eigener Vektor in der Interrupttabelle des Prozessors zugeordnet. Jeweils 4 Byte ab Adresse $0020 sind für IRQ0 bis 7 und ebenfalls jeweils 4 Byte ab Adresse $01C0 sind für IRQ8 bis 15.

2.2. Ansteuerung

Der Interruptcontroller 8259 löst in der üblichen PC-Einstellung einen Interrupt aus, wenn eine 1 (bzw. 5V) an einem der Interrupteingängen anliegt. Deshalb muß das zugehörige Interruptprogramm vor der Freigabe weiterer Interrupts erst dafür sorgen, daß der zugehörige Baustein die Leitung wieder zurückgesetzt. Wie im Teil 2 (1995/3-S.16) am Beispiel des Timers oder im Teil 4 (1995/2-S.29) mit RS232 geschildert, muß nach Bearbeitung der gewünschten Routine durch Einschreiben des Wertes $20 (unspezifischer **E**nd **O**f **I**nterrupt) in Portadresse $20 (Command-Register) der nächste Interrupt zugelassen werden.

Die Interruptleitung selbst wird durch Löschen des zugehörigen Bits im Maskenregister auf I/O-Adresse $21 aktiviert. Durch Setzen dieses Bits wird der Status für diese Leitung ignoriert und keine Interrupts mehr ausgelöst.

Bei den Interruptleitungen IRQ8 bis IRQ15 wird die Verkettung mit dem ersten Controller genutzt und über Eingang IRQ2 durchgeschleift. Es muß deshalb nach den Register des zweiten Interuptcontrollers auf Adresse $A0/$A1 auch der erste Controller wie oben beschrieben zurückgesetzt werden. Das Aktivieren der Interruptleitung IRQ2 wird normalerweise schon durch das Betriebssystem vorgenommen.

3. DMA-Ansteuerung

Sollen Datenmengen im Bereich von mehreren 10000Hz durch einen PC bearbeitet werden, so ist ein Interruptsystem nicht das geeignete Mittel. Schon allein durch den EMM386 ist ein Jittern (der Abweichung zwischen minimaler und maximaler Zeit) der Interruptbearbeitung von bis zu 50us zu beobachten. Außerdem kann jeder andere Interrupt das System beim Speichern eines Blockes auf Festplatte oder Diskette fast beliebig lange blockieren. Auch ein Maustreiber ist ein guter Kandidat für derartige Verzögerungen.

In solchen Fällen verwenden wir ähnlich wie z.B. der Festplattencontroller die am Steckplatz verfügbaren DMA-Leitungen. Diese sind mittels des eingebauten Controllers in der Lage, die Daten direkt von der Karte in einen angegebenen Speicher zu schreiben. Da dabei der Prozessor nicht eingreifen muß, braucht er nur für die Dauer der Übertragung angehalten werden.

Meist wird die DMA genutzt, um einen Datenblock schnell in den PC-Speicher zu bringen. Dazu wird bei jedem Signal von der Hardware ein Byte oder Wort (im AT) übertragen, Zieladresse erhöht und der DMA-Zähler erniedrigt. Bei 0 ist die Übertragung beendet.

In einem anderen Modus, der gut für die Meßdatenerfassung genutzt werden kann, verwendet man einen definierten Bereich im Speicher. Dieser wird wie zuvor mit Werten von der Meßkarte gefüllt. Hat der Zähler dann 0 erreicht, so springt der Adreßzeiger wieder auf den Anfang und der Zähler beginnt wieder mit dem Startwert. Dadurch kann ein Puffer von maximal 64KByte bzw. 128KByte bei 16Bit-DMA als Zwischenpuffer genutzt werden.

Hardwaremäßig hat man für jeden DMA-Kanal nur die beiden am Slot verfügbaren Leitungen DMA-Request (DRQ) und DMA-Acknowledge (-DACK). Um noch über Lesen und Schreiben zu entscheiden, werden dann noch -IOR bzw. -IOW benötigt. Will eine Slotkarte eine DMA-Übertragung auslösen, so setzt es die DRQ-Leitung. Sobald der DMA-Controller dann die Werte in den Speicher schreiben will oder Daten bereitstellt, zieht er die -DACK-Leitung auf 0 und steuert noch -IOR oder -IOW an. Die Adressen und das -MEMR bzw. -MEMW dürfen von der Karte nicht beachtet werden, weil damit gleichzeitig der Speicher angesteuert wird.

Tip: Bitte versuchen Sie niemals, ein Latch direkt über DMA einzulesen. Da sich der Wert noch in letzter „Sekunde“ ändern kann, liefert der Paritycontroller nicht den richtigen Wert. Dies führt dann beim Lesen der Daten aus dem Speicher zu einem Parityfehler, der meist mit erneutem Booten des Rechners geahndet wird.

3.1. Verwendung im PC

Im PC wurde nur ein DMA-Controller mit 4 Kanälen bestückt. Da es sich um einen 8Bit-Baustein handelte, kann er auch nur 8Bit-Daten und 16Bit-Adressen bearbeiten. Um die schon im ersten PC verfügbaren 20 Adreßleitungen nutzen zu können, wurde noch durch einen zusätzlichen Baustein die Adreßleitungen A16 bis A19 während der Übertragung auf vorgegebene Werte gesetzt. Leider hat diese geschichtliche Tatsache dazu geführt, daß in jedem PC der verwendete DMA-Puffer innerhalb eines Blockes von 64KByte begrenzt bleibt. Durch seine Beschaltung ist leider nur ein Übertragung von I/O-Bereich auf das RAM oder umgekehrt möglich. Dies ist aber kein Nachteil, da die heutigen Prozessoren sowieso schneller sind als die inzwischen zu langsame DMA.

Die netteste Möglichkeit, dem PC in den Selbstmord zu treiben, war die Abschaltung des DMA0. Sie wurde früher aus Kostengründen für den Refresh des RAM´s verwendet. Heute erledigen dies die großen namenlosen Chips auf dem Mainboard und der DMA0 kann frei genutzt werden. Die anderen Kanäle werden wie folgt verwendet:

Leitung	Verwendung
DMA0	früher RAM-Refresh, jetzt frei
DMA1	frei
DMA2	Diskette
DMA3	früher Harddisk, jetzt frei

3.2. Verwendung im AT

Natürlich wurde dann im AT das Seitenregister des ersten Controllers erweitert und kann jetzt A16 bis A23 ansteuern, was den Zugriff auf 16MByte erlaubt. Außerdem wurde ein zweiter, gleichartiger Baustein für DMA 4 bis 7 eingesetzt. Um aber für den erweiterten AT-Slot die 16Bit-Übertragung zu nutzen, wurde er speziell für diese Zweck anders betrieben. Durch Verschiebung der Adressen um ein Bit nach links (A0 ist bei 16Bit-Übertragung immer 0) kann der Baustein selbst 128KByte und mittels des ebenfalls verschobenen Seitenregister in manchen Rechnern bis zu 32MByte adressieren. Die Verwendung der zusätzlichen DMA-Kanäle ist wie folgt:

Leitung	Verwendung
DMA4	Verkettung mit ersten Baustein, nicht nutzbar
DMA5	PS/2-Hardisk, sonst frei
DMA6	frei
DMA7	frei

3.3. Programmierung

Der Ablauf der DMA-Programmierung geschieht wie folgt:

- Anforderung des DMA-Speichers (kein Seitenwechsel !!!)
- DMA abschalten
- Seitenregister setzen
- Anfangsadresse des DMA-Speichers im Controller setzen
- Größe des DMA-Speichers setzen
- Modus einstellen
- DMA aktivieren

- DMA auf der I/O-Karte initiieren
- Kontrolle, bis gewünschte Datenanzahl übertragen wurde
- DMA auf der I/O-Karte stoppen
- DMA-Kanal deaktivieren
- Daten verarbeiten
- Speicher freigeben

Dies nur theoretisch zu behandeln, ist sehr schwierig und wegen der vielen Register in den oft abgedruckten Listing manchmal falsch. Außerdem ist sowohl die Initialisierung der entsprechenden Hardware im PC-Slot als auch die Verarbeitung der Daten aus dem DMA-Puffer wichtig. Ich werde deshalb in einer späteren Folge eine konkrete Anwendung mit einer eigener Hardware schildern, weil es hier den Platz sprengen würde.

4. Wie geht's weiter

Hiermit findet die Serie zur PC-Meßtechnik seinen Abschluß. Jedoch ist deshalb keine Trauer angesagt, da die eigentliche Meßtechnik erst beginnt. Bei weitergehender Numerierung wird jetzt ohne notwendigen aber meist vorhandenen Bezug zum PC die Serie fortgesetzt. Dies geschieht, um mehr den Blick auf die eigentlichen Probleme (und natürlich auch deren Lösungen) bei Meß- und Steuerungsaufgaben freizuhaben. Neben dem PC werden deshalb auch vermehrt Beispiele zur Anwendung von Microcontroller aufgezeigt. Darüber hinaus sollen aber auch Hardware und Algorithmen nicht zu kurz kommen. Folgende Themen befinden sich dabei auf dem Stack:

- Informationen zu A/D und D/A-Wandler
- Preiswerte Sensoren für diverse physikalische Größen
- Integration und Differentiation
- Digitale Filter, Mittelungen und Glätten
- FFT für Signalanalyse und Synthese
- Beachtenswertes bei Regelkreisen
- Beispiele zum Einsatz von Meßtechnik in der Industrie

Bei allen Folgen wird dabei Wert auf die Praxis gelegt und meist eine entsprechende Anwendung geschildert oder sogar programmiert. Für den vertieften Einstieg in die Theorie muß deshalb auf die immer angegebene Literatur verwiesen werden.

Leider ist bei einem Erscheinungsintervall von 3 Monaten kaum an eine Beantwortung von Fragen über dieses Medium zu denken. Trotzdem würde ich mich über entsprechende Briefe oder Mails mit Fragen oder Anregungen freuen.

Literatur

(1) Thom Hogan
Die PC-Referenz für Programmierer
Microsoft Press

(2) Blank/Bernstein
PC-Schaltungstechnik in der Praxis
ISBN 3-89090-914-0
Markt-und-Technik-Verlag 1990

(3) Frank Van Gulluwe
The undocumented PC
ISBN 0-201-62277-7
Addison Wesley

(4) Tischler / Jennrich
PC INTERN (1-)5
ISBN 3-8158-1169-4
Data Becker GmbH&Co.KG (1995)