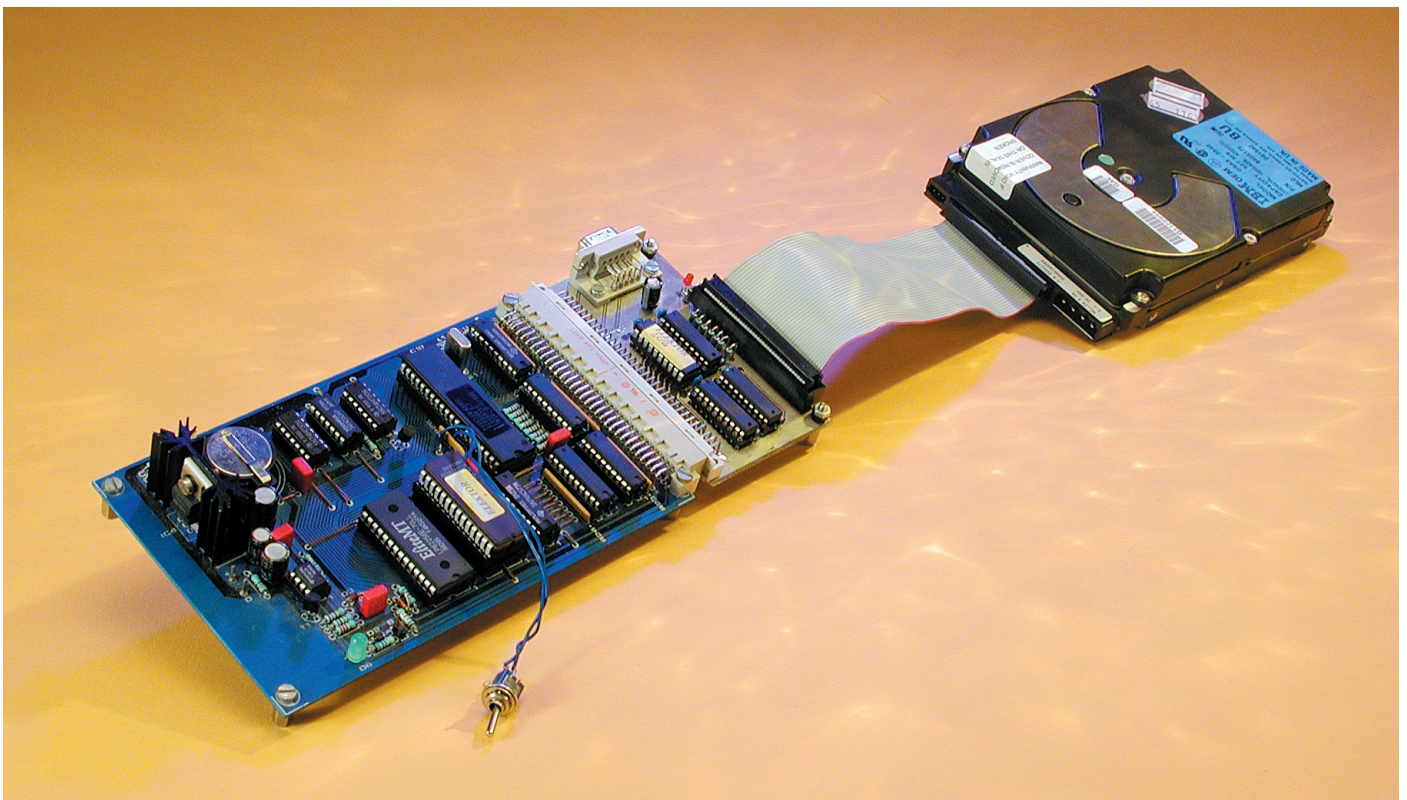


Harddisk-Interface für 8-bit-Controller

Controller sucht Festplattenanschluss

Von Stefan Schwark

Obwohl die IDE-Schnittstelle auch für 8-bit-Betrieb ausgerüstet ist, unterstützen die meisten Festplatten diesen Modus nicht. Will man sie dennoch mit einem 8-bit-Mikrocontroller steuern, kommt man um das hier beschriebene Interface nicht herum.



Festplatten als Massenspeicher sind aus den heutigen Computern nicht mehr wegzudenken. Waren sie noch vor ein paar Jahren grosse und teure Geräte, so sind sie heute mit (bis zur nächsten Generation) riesigen Kapa-

zitäten und kleinen Abmessungen recht preisgünstig zu bekommen. Die meisten Computer benutzen heute Festplatten mit IDE-Interface. Da diese auch ATA-Interface

genannte Schnittstelle aus dem PC-Bereich stammt und somit mit einem 16-Bit-Bussystem arbeitet, ist ein direkter Betrieb mit 8-bit-Controllern möglich, wenn auch problematisch.

Denn obwohl die Schnittstelle in den 8-bit-Betrieb umgeschaltet werden kann, funktioniert das in der Praxis nicht immer, da dieses Feature von vielen Festplatten nicht unterstützt wird. Man ist also gezwungen, in der Mikrocontroller-Schaltung einen 16-Bit-Datenbus anzubieten. Genau das ist die Aufgabe des hier beschriebenen Interface. Es wurde entwickelt, um mit dem 1991 in Elektor veröffentlichten 8032-Compu-board die weit verbreiteten IDE-Festplatten ansteuern zu können. Aufgrund des relativ unkomplizierten Aufbaus lässt es sich aber ohne grosse Probleme auch an anderen Mikroprozessoren verwenden.

Das *Integrated Disk Electronic Interface*, kurz IDE-Interface wurde 1984 von der Firma Compaq entwickelt. Das Ziel war, den Festplattencontroller (bis dahin eine ziemlich grosse Steckkarte) in die Festplatte zu integrieren. Die PC-Steckkarte konnte jetzt viel einfacher ausfallen, da sie nur noch für die Verbindung mit dem Systembus gebraucht wurde. Der IDE-Anschluss ist im Prinzip nichts weiter als ein auf die für die Festplatte benötigten Leitungen reduzierter AT-Slot. Deswegen heisst das Interface korrekt auch ATA-Interface (AT-Attachment). Im PC wird die Festplatte über die Adressen 1F0...1F7h und 3F0...3F7h angesprochen. Es gibt zwei CS-Leitungen (**C**ard **S**elect), die die einzelnen Bereiche in der Festplatte auswählen. Die eigentliche Registerauswahl erfolgt über drei Adressleitungen. Im hier vorgestellten Interface werden andere Adressen verwendet. Alle beiden Registerbereiche liegen im Adressbereich C000h bis C00Fh. Die einzelnen Register und ihre Bedeutung sind im Abschnitt über die Software aufgeführt.

Um die Festplatte anzusprechen, bedient man sich des ATA-Befehlsatzes. Hier sind die Codes definiert, die die verschiedenen Funktionen auslösen (siehe **Tabelle 1**). Die Befehle an die Festplatte werden in das Kommandoregister an Adresse 1F7h (entspricht C00Fh) geschrieben. Die einzelnen Kommandos sind in der Spezifikation des ATA-Befehlsatzes festgelegt. Es gibt aber auch hier herstellerbedingte Unterschiede, so dass nicht jede Festplatte jedes Kommando versteht.

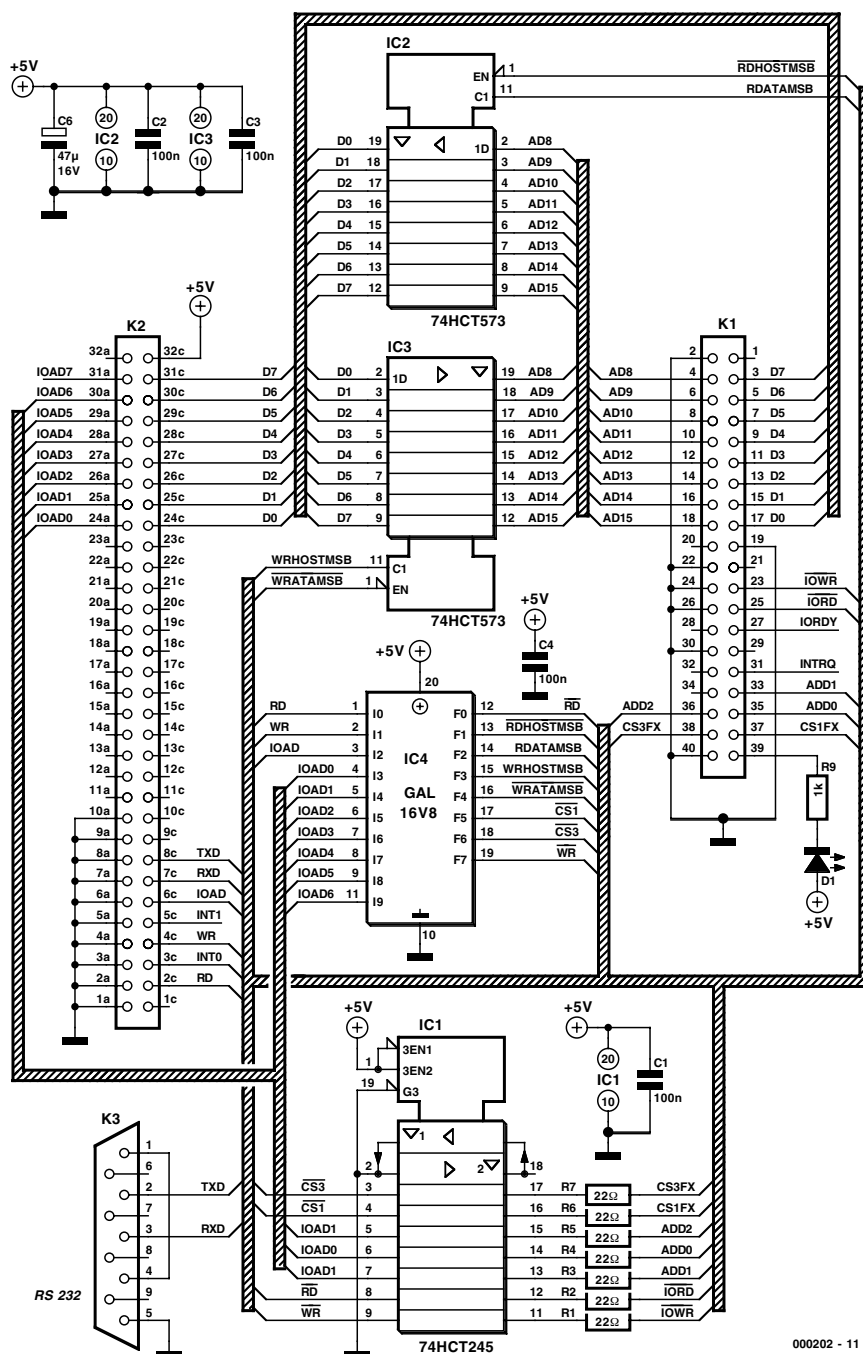


Bild 1. Herzstück des Interfaces ist der GAL-Baustein IC4.

Seit seiner Entwicklung hat das ATA Interface eine beachtliche Verbreitung erreicht und ist heute in den meisten PCs das Interface für Festplatte und CD-ROM. Da die ursprüngliche Spezifikation für Geräte wie CD-ROM-Laufwerke ungeeignet war, hat man in den letzten Jahren eine Erweiterung eingeführt. Diese ATAPI Schnittstelle (ATA Packet Interface) entspricht auf der Hardwareseite voll dem ATA-Inter-

face. Die Transfermodi und das Registermodell sind gleich. Während die normalen ATA-Befehle meistens nur aus einem Kommando bestehen, der in das Kommandoregister geschrieben wird, verwendet ATAPI ein 12 byte langes Kommandopakete. Diese Datenpakete orientieren sich im Aufbau stark an SCSI, und so ist es verständlich, dass zum Beispiel IDE-CD-ROM-Laufwerke und Tape-streamer (ZIP) in Wirklichkeit SCSI-Geräte sind, die ihre Kommunikation über eine Art Übersetzungsstufe abwickeln.

Stückliste

Widerstände:

R1...R7 = 22 Ω
R8 = 1 k

Kondensatoren:

C1...C4 = 100 n
C5 = 47 μ /16 V

Halbleiter:

D1 = High-efficiency-LED, rot
IC1 = 74HCT245
IC2, IC3 = 74HCT573
IC4 = 16V8 (programmiert)

Außerdem:

K1 = 2x20-poliger Pfostenverbinder mit Schutzkragen und Orientierungsnase
K2 = 64-poliger Verbinder DIN41612-C (female, gewinkelt)
Platine EPS 000202-1
Software EPS 000202-11 (C-Beispielprogramm und GAL-Listing)
Prog. GAL EPS 000202-31

Die Hardware

Die Hardware des Interfaces kommt mit ganzen vier ICs aus. Das Herz der Schaltung in **Bild 1** ist das GAL16V8 (IC4) zur Ansteuerung der Latches IC2 und IC3. Der Quellcode des GAL-Programms kann kostenlos von der Elektor-Downloadsite herunter geladen werden, er ist aber auch auf Diskette (EPS 000202-11) beim Verlag erhältlich. Es generiert einige der für das ATA-Interface erforderlichen Steuersignale, so die beiden Card-select-Signale CS1Fx und CS3Fx. Diese beiden Signale selektieren die zwei Registerbereiche, die in der Platine zur Verfügung stehen. Die Bezeichnung der beiden Signale stammt aus dem PC-Bereich, da die Register dort ja an Adresse 1F0..1F7h beziehungsweise 3F0..3F7h gefunden werden. Das GAL des Interfaces verwendet den Adressbereich C000..C00Fh, in den alle Register eingeblendet werden. Die Auswahl der einzelnen Register erfolgt über die Adressleitungen A0...A2. Eine Auflistung der einzelnen Register ist im Abschnitt über die Software zu finden.

Auch die beiden Signale $\overline{WR}$ und $\overline{RD}$ werden vom GAL erzeugt. Hierzu werden die Controllersignale RD und WR invertiert. Die übrigen Ausgänge steuern IC2 und IC3. IC2 wird zum Zwischenspeichern des MSB gebraucht, wenn der Prozessor ein Datenwort vom Bus liest. Sobald der Host einen Leseauftrag über das ATA-Interface schickt, lädt IC2 das MSB von den Datenleitungen D8...D15 in seinen Speicher. Wenn der Prozessor jetzt die

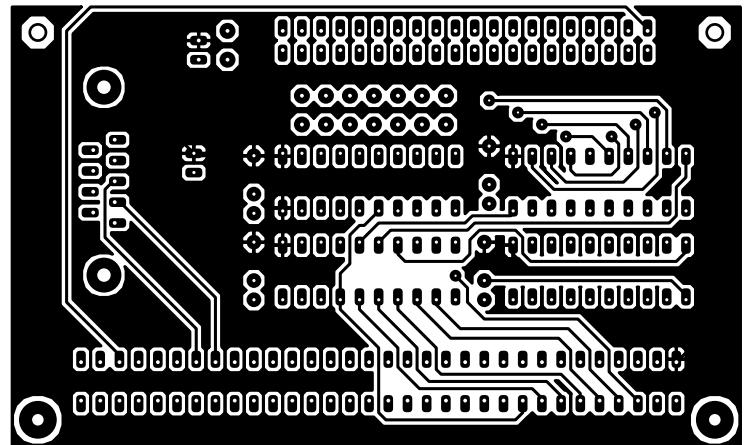
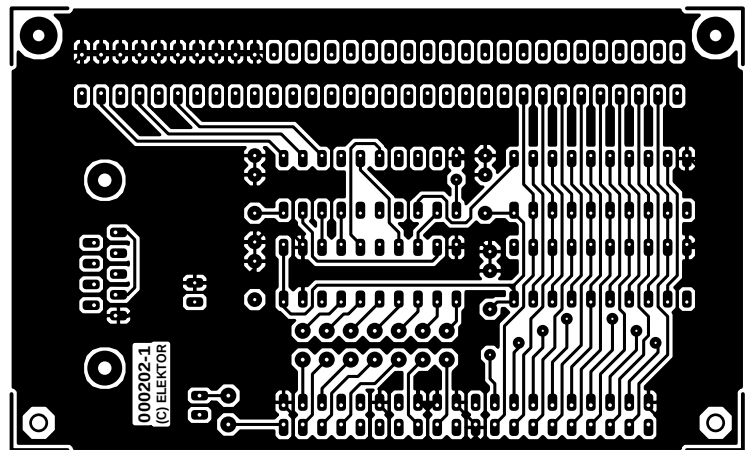
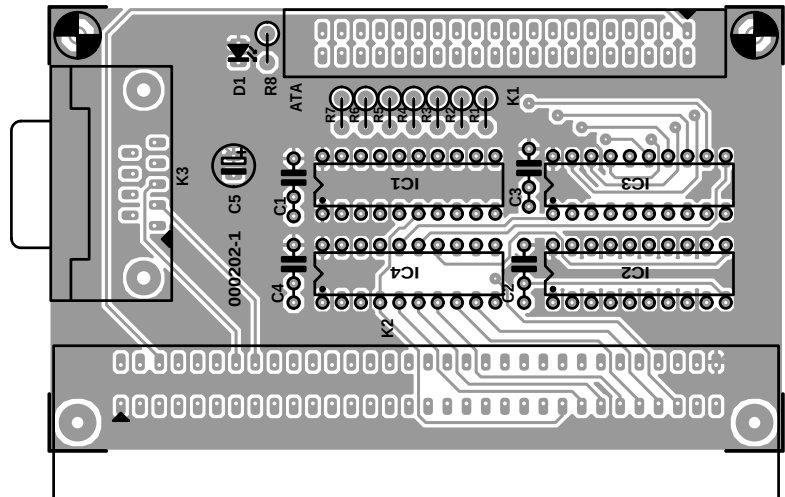


Bild 2. Platinenlayout und Bestückungsplan der kleinen Interface-Platine.

Adresse C000h liest, gibt IC2 das letzte gelesene MSB über den Datenbus an den Prozessor weiter. IC3 übernimmt diese Aufgabe bei einem Schreibauftrag an das ATA-Interface. Der Host schreibt das betreffende MSB nach Adresse C000h wobei IC3 dieses Byte in seinen Speicher über-

nimmt. Wenn die Festplatte die Daten lesen will, legt das IC das MSB auf die Datenleitungen D8...D15. Das letzte verbleibende IC1 wird gebraucht, um die Steuersignale, die IC4 und den Prozessor mit dem ATA-Interface verbinden, zwischenzupuffern.

Tabelle 1. ATA-Befehle

Kommando	Kode	FR	SC	SN	CY	DH
CHECK POWER MODE	E5h		x			D
EXECUTE DEVICE DIAGNOSTIC	90h					D
IDENTIFY DEVICE	Ech					D
IDLE	E3h		x			D
IDLE IMMEDIATE	E1h					D
INITIALIZE DEVICE PARAMETERS	91h		x			x
READ MULTIPLE	C4h		x	x	x	x
READ SECTOR(S)	20h 21h		x	x	x	x
READ VERIFY SECTOR(S)	40h 41h		x	x	x	x
SEEK	70h			x	x	x
SET FEATURES	Efh	x				D
SET MULTIPLE MODE	C6h		x			D
SLEEP	E6h					D
STANDBY	E2h		x			D
STANDBY IMMEDIATE	E0h					D
WRITE DMA	CAh CBh		x	x	x	x
WRITE MULTIPLE	C5h		x	x	x	x
WRITE SECTOR(S)	30h 31h		x	x	x	x

FR = Feature Register C009h
SC = Sektorzähler C00Ah
SN = Sektor Nummer Register C00Bh
CY = Zylinder Register C00Ch und C00Dh
DH = Kopfnummerregister
Hier bedeutet ein D, dass nur die Deviceparameter des Registers von Bedeutung sind.

Auf der Platine ist neben dem 40-poligen ATA-Verbinder K1 und dem 64-poligen DIN4162-C-Verbinder zum Anschluss des Mikrocontrollerboards auch ein 9-poliger Sub-D-Stecker zu finden. Auf diesen werden die seriellen Leitungen des Controllerboards durchgeschleift. C1...C5 sind die üblichen Abblockkondensatoren, LED D1 zeigt die Laufwerksaktivität an, wenn die aktive Platte diese Leitung gegen Masse zieht. Eine Unterscheidung über Master- oder Slave-Platte ist nicht möglich,

da beide Plattensignale den gleichen Pin benutzen.

Der Aufbau der Interface-Platine (Bild 2) gestaltet sich ohne Komplikationen. Die ICs können mit Fassungen ausgestattet werden. Achten Sie bei den Lötarbeiten auf die richtige Polarität von C6, der LED und natürlich auch der ICs.

Die Software

Um das ATA-Interface von einem Mikrocontroller direkt anzusteuern,

braucht man ausser der notwendigen Hardware natürlich noch die passende Software, die die einzelnen Kommandos in die Festplattenregister schreiben muss. Die Register der Festplatte sind weiter unten beschrieben. Unter der hinter dem Registernamen angegebenen Adresse wird das ATA-Interface vom Compuboard angesprochen. Einige der Register haben eine andere Bedeutung, wenn die Festplatte im LBA-Modus (Logical Block Addressing) betrieben wird. Die Adressierung der einzelnen Sektoren geschieht hierbei nicht mehr nach Zylinder, Kopf und Sektor, sondern die einzelnen Festplattensektoren werden linear durchnummeriert. LBA wurde eingeführt, um die für das ATA-Interface geltende 504-MB-Kapazitätsgrenze aufzuheben. (Manchmal findet man auch 528MB. Dann ist für ein KByte mit 1000Byte gerechnet worden). Der erste Sektor besitzt die Adresse 0. Bei einer 540-MB-Platte hätte der letzte physikalische Sektor dann die Nummer 1.065.456. Durch diese Art der Adressierung konnte die Grenze auf 7,8 GB verschoben werden. Wer also eine Platte mit mehr als 504 MB anschliessen möchte, muss die Platte in der LBA-Betriebsart betreiben.

Datenregister C000h

Die Ein- und Ausgabe der Datenbits 8...15 erfolgt über dieses Register.

Datenregister C008h

Hier werden die Datenbits 0...7 übertragen. Die Daten sind gültig, wenn im Statusregister das DRQ-Bit gesetzt ist.

Fehlerbits/Feature C009h

Im Lesezugriff bekommt man hier die Fehlerbits geliefert, anhand derer man die Art des aufgetretenen Fehler erkennen kann. Im Schreibzugriff werden hier Parameter zum Beispiel für Cache On/Off übergeben.

Sektorzähler C00Ah

Dieses Register enthält die Anzahl der mit dem nächsten Laufwerkszugriff zu bearbeitenden Sektoren.

Die Bedeutung ist für Schreib- und Lesezugriff gleich.

Startsektor C00Bh

Hier kann der Startsektor für den nächsten Zugriff geschrieben oder gelesen werden. Im LBA-Mode sind hier die LBA Bits 0...7 zu finden.

Zylinder LSB C00Ch

Unter dieser Adresse stehen die unteren 8 bit der Zylinderadresse, im LBA-Mode die LBA-Bits 8...15.

Zylinder MSB C00Dh

Das sind die oberen 8 bit der Zylinderadresse

Tabelle 2. Einträge der Partitionstabelle

Offset-Adresse	Inhalt
00h	Partitionsstatus (00h=inaktiv, 80h=Boot-Partition)
01h	Schreib-/Lesekopf, mit dem die Partition beginnt
02h	Sektor und Zylinder, mit dem die Partition beginnt
04h	Partitionstyp (z.B. 04h für DOS 16 Bit FAT)
05h	Schreib-/Lesekopf, mit dem die Partition endet
06h	Sektor und Zylinder, mit dem die Partition endet
08h	Entfernung des ersten Sektors der Partition (Boot-Sektor) vom Partitionssektor in Sektoren
0Ch	Anzahl der Sektoren in dieser Partition

oder die LBA-Bits 16...23.

Laufwerks-, Kopfnummer C00Eh

Die Bits 0...3 enthalten die binär kodierte Kopfnummer. Im normalen Modus gibt es maximal 16 Köpfe. Bit 4 dient zur Unterscheidung von Master und Slave (1 = Slave, 0 = Master). In den Bits 5...7 steht die Anzahl der Bytes/Sektor. Hier ist 512 Bytes/Sektor fest eingetragen. Im LBA-Mode sind hier die LBA-Bits 24...27 zu finden.

Status-/Kommandoregister C00Fh

In das Register werden die ATA-Kommandos geschrieben. Im Lesezugriff ist dies das Statusregister. An den einzelnen Bits kann der aktuelle Zustand der Platte erkannt werden:

- **Bit 0 (ERR)** zeigt einen Fehler an. Die Art des Fehlers kann über die Fehlerbits an Adresse 1F1h ermittelt werden.
- **Bit 1 (IDX)** ist ein herstellerabhängiger Indeximpuls.
- **Bit 2 (CORR)** zeigt die Übertragung von mit ECC korrigierter Daten an.
- **Bit 3 (DRO)** Das Gerät ist bereit zum Datentransfer.
- **Bit 4 (DSC)** Der Suchvorgang ist fertig ausgeführt.
- **Bit 5 (DF)** Es wurde ein Schreibfehler festgestellt.
- **Bit 6 (DRDY)** Das Laufwerk zum Beispiel nach dem Einschalten bereit.
- **Bit 7 (BUSY)** Das Laufwerk ist mit der Ausführung eines Befehls beschäftigt.

2. Statusregister/Device Control C006h

Der Lesezugriff liefert das Gleiche wie bei einem Zugriff auf das erste Statusregister. Der Zustand der Interrupt-Leitung wird aber nicht verändert. Bei einem Schreibzugriff kann hier durch Setzen von Bit 1 auf 0 der Interrupt aktiviert werden. Wird Bit 2 auf 1 gesetzt, führt das zu einem Software-Reset der Festplatte. Die Bits 3...7 werden nicht benutzt. Bit 0 muss immer 0 sein.

Aktive Adresse C007h

Dieses Register kann nur gelesen werden. Bit 0 und Bit 1 unterscheiden Master- und Slave-Laufwerk. In den Bits 2...5 findet man das Einerkomplement des eingeschalteten Kopfes. Eine Schreibaktivität des Laufwerks wird durch eine 0 in Bit 6 angezeigt. Bit 7 zeigt beim PC an, dass die Diskette aus dem Floppy-Laufwerk entnommen wurde. Für unsere Zwecke ist dieses Bit nicht benutzt.

Um das Programmieren etwas einfacher zu gestalten, hat der Autor ein kleines C-Programm (ebenfalls im Internet und auf Diskette) geschrieben, um die Festplattenansteuerung zu demonstrieren. ATA.C initiali-

Tabelle 3. Belegung ATA-Interface (KI)

Pin	Signal	Beschreibung
2,19,22,24,26,30,40	GND	Masse
1	RES	Reset-Signal vom Rechner
17	D0	Bidirektionaler Datenbus zwischen Festplatte und Rechner. Über diesen Datenbus werden Daten-, Status- und Steuerinformationen übertragen. Wenn das Laufwerk nicht selektiert ist, werden die Leitungen hochohmig geschaltet.
15	D1	
13	D2	
11	D3	
9	D4	
7	D5	
5	D6	
3	D7	
4	D8	
6	D9	
8	D10	
10	D11	
12	D12	
14	D13	
16	D14	
18	D15	
20		Abgekniffener Pin als Schutz gegen Verdrehen des Kabels.
21	/IOCHRDY	Der Rechner muss bei Laufwerkszugriffen warten, bis dieses Signal inaktiv ist.
23	/IOWR	Schreibsignal für E/A-Portadressen
25	/IORD	Lesesignal für E/A-Portadressen
27	/IOCHRDY	wie Pin 21
28	ALE	Adress Latch Enable
29	frei	reserviert für Erweiterungen
31	IRQ14	Unterbrechungsanforderung an den Rechner.
32	IO16	Anzeige für 16 Bit Datentransfer
34	/PDIAG	Passed Diagnostics: Das Slave Laufwerk meldet an den Master das die interne Diagnose abgeschlossen ist.
33	A1	Adressleitungen vom Rechner, über die die einzelnen Register ausgewählt werden.
35	A0	
36	A2	
37	/CS0	Adressen 1F0h bis 1F7h
38	/CS1	Adressen 3F0h bis 3F7h
39	/ACT	Kann eine LED treiben, um die Laufwerksaktivität anzuzeigen.

siert die serielle Schnittstelle, da später hierüber Meldungen ausgegeben werden sollen. Anschliessend wird ein Reset des Interfaces ausgelöst. Wenn die Platte sich dann bereit meldet, schickt das Programm ein IDENTIFY-DEVICE-Kommando an die Festplatte. Die Festplatte beginnt jetzt, ihre internen Daten auszugeben. Im Einzelnen sind das

Daten über Modell, Version, Seriennummer, Anzahl der Köpfe und Zylinder und so weiter. Die von der Festplatte gesendeten Informationen werden über die serielle Schnittstelle ausgegeben. Jetzt wird der erste Datenblock des Bootsektors gelesen und ebenfalls ausgegeben. Hierbei ist wichtig, das immer ganze 512 Byte ausgelesen werden, da es sonst

bei späteren Lesevorgängen zu Problemen kommen kann. Dieses kleine Beispiel zeigt recht gut, wie einfach es ist, eine IDE-Festplatte anzusprechen. Es sind auch einige Routinen in MCS51-Assembler verfügbar, die in selbst geschriebene Programme übernommen werden können.

Einteilung der Festplatte

Um die Daten, die auf die Festplatte geschrieben werden, wieder zu finden, braucht man natürlich noch ein System. Die erste Stufe der Festplatteneinteilung ist die so genannte Low-Level Formatierung. Hierbei werden auf der Oberfläche der Festplatte Spuren mit mehreren Sektoren festgelegt. Jeder Sektor ist 512 Bytes gross. Nach dieser Low-Level-Formatierung ist die Festplatte wie eine riesige Diskette. Für einige Anwendungen wäre dieser Zustand jetzt schon ausreichend. Wenn man etwa nur Daten über längere Zeit aufzeichnen will, könnte man auch einen Sektor nach dem anderen mit 512 Byte grossen Datenpaketen beschreiben. Normalerweise wird man aber mit dieser sequentiellen Nutzung nicht auskommen.

Als normaler PC-Anwender braucht man sich um die genaue Funktion der Festplatte nicht zu kümmern. Das übernehmen alles die Gerätetreiber des Betriebssystems oder das BIOS. Will man allerdings mit selbst geschriebener Software auf eine Platte zugreifen, kommt man um die internen Strukturen der Platte nicht herum.

Eine Festplatte umfasst eine oder mehrere Partitionen. Diese Partitionen (auch Volumes genannt) sind verschiedene genau abgegrenzte Bereiche der Festplatte. Der Sinn ist die Bereitstellung verschiedener Festplattenbereiche, die von unterschiedlichen Betriebssystemen verwaltet werden können.

Nach der Ausführung des Speichertests, dem Test verschiedener Grundfunktionen des Motherboards und dem initialisieren der Grafikkarte lädt das BIOS den ersten Sektor einer Festplatte (Kopf 0, Zylinder 0, Sektor 1), den Partitionssektor. Wenn das BIOS in den letzten beiden der 512 Bytes die Codefolge 55AAh findet, so erkennt es den Sektor als ausführbaren Code und startet mit

dem ersten Sektorbyte die Programmausführung an der Adresse 0000:7C00. Der hier gespeicherte Code hat die Aufgabe, die aktive Partition zu erkennen und den dort vorhandenen Bootsektor auszuführen. Wo dieser Sektor zu finden ist, kann die Routine am Inhalt der Partitionstabelle (normalerweise ab Offset 01BEh im Partitionssektor) erkennen. Diese Tabelle enthält für jede Partition einen Eintrag von 16 Bytes. Welche dieser Partitionen die aktive Bootpartition ist, kann die Routine am ersten Eintrag feststellen, der bei einer aktiven Partition 80h, bei einer inaktiven 00h ist. Wenn keine oder mehrere Partitionen aktiviert sind, bricht die Routine die Ausführung ab.

Die Bedeutung eines Eintrags der Partitionstabelle ist in **Tabelle 2** aufgelistet. Diese Einträge sind zwar unverbindlich, aber praktisch bei jedem Betriebssystem vorhanden. Das Betriebssystem kann mit der Partition natürlich noch nicht viel anfangen. Der Bereich muss jetzt erst noch formatiert werden. Hierbei werden die physikalischen Sektoren in logische Bereiche (Cluster) aufgeteilt, die das jeweilige Betriebssystem dann zusammenhängend nutzen kann. Die Verwaltung dieser Cluster hängt vom benutzten Betriebssystem ab. Bei DOS nutzt man die *File Allocation Table (FAT)*. Jeder Eintrag in dieser Datenstruktur bezeichnet einen dieser Cluster. Hier steht etwa, zu welcher Datei dieser Cluster gehört und in welchem Cluster die Datei fortgeführt wird. Die Nutzung dieser FAT beim Lesen und Schreiben von Dateien muss der Literatur zum verwendeten Betriebssystem entnommen werden, da das Thema den Rahmen dieses Artikels sprengen würde.

Der Inhalt von Byte 04h bezeichnet das Betriebssystem:

Byte 04h	Betriebssystem
00h	Nicht gebrauchte Partition
01h	DOS 12-bit-FAT
04h	DOS 16-bit-FAT
05h	Erweiterte Partition
08h	OS/2
0Bh	FAT-32
41h	PowerPC Boot
51h	Novell
64...69h	Novell
81h	Linux

Tabelle 4. Speichereinteilung des ATA-Interface		
Adresse	Lesen	Schreiben
C000h	D8...D15	
C006h	alternate status	device control
C007h	drive address	
C008h	Data	
C009h	error	Feature
C00Ah	sector count	
C00Bh	sector number	
C00Ch	cylinder low	
C00Dh	cylinder high	
C00Eh	drive/head	
C00Fh	staus	Command
Statusregister		
D7	BUSY	device busy
D6	DRDY	device ready
D5	DF	devive fault
D4	DSC	device seek complete
D3	DRQ	data request
D2	CORR	correction
D1	IDX	vendor specific
D0	ERR	error
Device control register		
D7...D23		reseved
D2	SPST	soft reset
D1	NIEN	interrupt enable (active low)
D0	0	always 0

83h	Linux swap
A5h	FreeBSD
A9h	NetBSD

Die Möglichkeiten dieses Interface sind mit dem Anschluss einer Festplatte noch nicht ausgereizt. Da die ATAPI-CDROM Schnittstelle das gleiche Registermodell verwendet, lässt sich die vorgestellte Schaltung auch hierfür einsetzen. Für diese ATAPI-Geräte gibt es eine Reihe Spezialbefehle, über die sich die normalen Abspielfunktionen recht einfach ansprechen lassen. Vielleicht kommt das alte Double-Speed-CDROM aus der Bastelkiste ja doch noch mal zum Einsatz?

(000202)rg

Internetadressen mit Informationen zum Thema finden Sie auf der Internetseite des für die Spezifikation des ATA Standards zuständigen T13 Komitee

<http://www.t13.org>