Nicht nur ein Geheimdienst: CIA

Berühmt und berüchtigt. Häufig defekt und schwer zu beschaffen. Das sind die beiden 6526-Portbausteine des C 64. Aber wissen Sie auch, welche ungeahnten Möglichkeiten in diesen Bausteinen stecken?

Im C 64 finden sich zwei ICs, über welche die CPU, die Hauptsteuereinheit des C 64, in Kontakt mit der Außenwelt tritt. Tastatur- und Joystickabfrage wären ohne diese Ein-/Ausgabe-Bausteine nicht möglich. Auch der serielle Bus, die Schnittstelle für Drucker und Diskettenlaufwerke, ist auf sie angewiesen.

Die im C 64 eingebauten Port-Bausteine stellen eine technische Weiterentwicklung der VIAs (Versatile Interface Adapter) 6522 dar, die im »kleinen Bruder« des C 64, dem VC 20, zur Verwendung kamen. Sie tragen den leicht zu Verwechslungen führenden Namen »CIA«, was aber hier für »Complex Interface Adapter« und nicht für »Central Intelligence Agency« steht. Die Typenbezeichnung dieser Bausteine ist 6526. Das weist darauf hin, daß diese Chips zum Anschluß an die Prozessoren der 65xx-Familie gedacht sind. Im C 64 ist der Prozessor ein 6510, im VC 20 ein 6502.

Einige unter Ihnen werden schon schlechte Erfahrungen mit den 6526-Bausteinen gemacht haben. Häufig löst sich nämlich die Funktionswilligkeit der CIAs in einem »Amperewölkchen« auf. Meist dann, wenn bei eingeschaltetem C 64 Zusatzplatinen an die Ports angeschlossen oder von diesen abgezogen werden. Tödlich für die CIAs ist dabei die statische Elektrizität, mit der man sich beispielsweise auf Teppichboden leicht auflädt und die dann über die CIA abfließt. Weiß man, welche CIA defekt ist, kommt das Beschaffungsproblem: Commodore gibt die Bausteine nur an Händler ab und die wollen verständlicherweise den Computer selbst reparieren.

Öffnet man das Gehäuse des Computers, so sieht man in der linken oberen Ecke zwei 40polige ICs. An dieser Stelle eine Warnung an alle, die ihren Computer noch nicht länger als sechs Monate haben: Durch das Öffnen des Gehäuses kann jeder Garantieanspruch verlorengehen! Schauen wir uns zunächst die allgemeinen Eigenschaften der CIA 6526 an. Sie besitzt zwei 8-Bit-Parallel-Ports mit den dazugehörigen Handshake-Leitungen, zwei programmierbare 16-Bit-Zähler (Timer), eine 24-Stunden Echtzeituhr mit einer Auflösung von einer Zehntelsekunde und eine serielle Ein-/Ausgabe-Schnittstelle.

Eine CIA hat 16 8-Bit-Register, die für die Steuerung der einzelnen Funktionen zuständig sind. Sie können, wie der RAM-Speicher, mit PEEK und POKE angesprochen werden. Wichtig ist hierbei auch, daß die CIAs direkt mit dem Adreßbus des Prozessors verbunden sind. Die erste Adresse der ersten CIA liegt bei 56320 ($DC00),die der zweiten bei 56576 ($DD00). Will man eines der Register ansprechen, so addiert man zu der Basisadresse einfach die Nummer des Registers (0 bis 15) hinzu. Die Inhalte der einzelnen Register wiederholen sich dann noch 15mal. So ist auch der große Abstand der Basisadressen von CIA 1 und CIA 2 zu erklären. Für die Hardware-Freaks sei hier kurz die Ursache dafür erwähnt: Die CIAs besitzen nur vier Adreßleitungen, die mit den vier niederwertigen Bits des Adreßbusses verbunden sind. Es muß ihnen also über das Chip-Select-Signal (Pin 23, Bild 1) mitgeteilt werden, wann der Prozessor mit ihnen in Verbindung treten will. Da aber das CS-Signal für die betreffende CIA aufgrund der Adreßdecodierung in der ganzen Page, also von $DC00 bis $DCFF beziehungsweise $DD00 bis $DDFF, aktiviert ist, fühlt sich die CIA im Bereich der ganzen Page angesprochen. Zur Registerauswahl werden jedoch nur die unteren vier Bits benutzt.

Die CIA 6526 besitzt zwei voneinander unabhängige 8-Bit-Ports (Schnittstellen), mit denen der C 64 Daten mit seiner Umwelt über den User-Port austauschen kann. Für jeden dieser Ports existiert ein Register, welches die Zustände der einzelnen Port-Leitungen (Port A: Pin 2 bis 9, Port B: Pin 10 bis 17) Bit für Bit widerspiegelt. Liegt an einer dieser Leitungen Spannung an (High-Pegel), so ist das entsprechende Bit im Datenregister gesetzt. Liegt am Pin keine Spannung an (Low-Pegel), so ist das Bit in diesem Register gelöscht. Dieses Register heißt Datenregister. Um aber auch einen, wie der Fachmann sagt, »bidirektionalen« Datenverkehr zu ermöglichen, gibt es zu jedem Port neben dem Datenregister noch ein Datenrichtungsregister (Port A: Register 2, Port B: Register 3).

Die CIA — ein Wunderwerk an Funktionen

Bidirektionaler Datenaustausch heißt, daß der Computer sowohl Daten über den CIA-Port, der am User-Port herausgeführt ist, ausgeben als auch empfangen kann. Ein großer Vorteil liegt darin, daß man hiermit die Port-Leitungen einzeln und unabhängig voneinander als Ein- oder Ausgänge schalten kann.

Ist ein Bit im Datenrichtungsregister eines Ports gelöscht, so arbeitet das entsprechende Port-Bit als Eingang. Ein solcher Ausgang besitzt über einen hochohmigen Pull-up-Widerstand logischen High-Pegel. Um ein Port-Bit als Ausgang (+5V) zu definieren, muß das entsprechende Datenrichtungsregister-Bit gesetzt werden. Das erklärt auch, warum man keinesfalls einen als Eingang geschalteten Port zu einem Ausgang machen darf. Denn angenommen, ein Port-Eingang läge auf 0V, dann fließt über den Pull-up-Widerstand ein kleiner, unbedeutender Strom, und die Welt ist für den C 64 in Ordnung. Das kann sich aber schnell ändern, wenn der Eingang über das Datenrichtungsregister als Ausgang (High-Pegel) geschaltet wird. Denn dann ist ein Kurzschluß unvermeidlich, da die 5V-Spannung jetzt direkt an Masse liegt und nicht, wie im Normalfall, über einen Widerstand.

Man kann also einem als Ausgang programmierten Port durch POKEn eines geeigneten Wertes in das Datenregister einen bestimmten elektrischen Zustand geben. Bei jedem Zugriff auf dieses Datenregister, egal ob PEEK oder POKE, erscheint am Pin PC (Pin 18 der CIA) ein kurzer Impuls, der dazu verwendet werden kann, dem Partner beim Datenaustausch, einem zweiten C 64 oder einem Drucker, mitzuteilen, daß Daten empfangen oder gesendet werden. Der Impuls dauert einen Systemtakt (etwa 1 Mikrosekunde). Der Pin PC von CIA 2 liegt am User-Port an Anschluß 8.

Die Ports der CIA werden zum Beispiel zur Datenübermittlung mit anderen Computern oder Peripheriegeräten verwendet. Möglich wäre auch eine Centronics-Schnittstelle, mit der man auch andere Drucker als Commodore-Drucker anschließen kann. Da der CIA-Port acht Bit breit ist, lassen sich immer 8 Bit (1 Byte) gleichzeitig übertragen. Man spricht dann von einer 8-Bit-Parallelübertragung.

Man kann mit dem User-Port aber auch Roboter steuern oder durch den C 64 als Alarmanlage das Haus überwachen lassen. Der Phantasie sind hier praktisch keine Grenzen gesetzt. Wir wollen aber nicht noch näher auf den User-Port eingehen, denn er wurde im 64'er, Ausgabe 5/85, Seite 36 schon eingehend besprochen.

Eingebaute Uhren: Die Timer

Wie bereits erwähnt, besitzt die CIA 6526 zwei programmierbare Timer. Das sind Zähler, die einen wählbaren Wert bis 0 dekrementieren (herunterzählen). Der Wert kann maximal 16 Bit groß sein, also 65535. Timer A belegt Register 4 (Low-Byte) und Register 5 (High-Byte), Timer B die Register 6 und 7. Auf einen Impuls hin, den der Fachmann »Trigger« nennt, erniedrigt der Timer den eingestellten Wert um 1. Der Trigger-Impuls kann für beide Timer von verschiedenen Quellen kommen. Es kann entweder der Systemtakt oder ein positives Signal am Pin CNT sein. »CNT« steht für »Count«. Dieser Pin ist für beide CIAs am User-Port herausgeführt (Pin 4 für CIA 1 und Pin 6 für CIA 2). Außerdem kann Timer B von Timer A getriggert werden, nämlich jedesmal, wenn dieser den Wert Null erreicht hat. Dadurch ist es möglich, aus den zwei Timern einen einzigen zu machen, der 32 Bit umfaßt, also von 0 bis 2^{32}-1 (= 4.294.967.295) zählen kann.

Ein Erreichen negativer Werte nennt man Unterlauf. Bei jedem Unterlauf sieht der Timer in seinem Kontrollregister nach (Timer A: Register 14, Timer B: Register 15), ob das Bit 3 gesetzt ist. Wenn das der Fall ist, bleibt der Timer stehen. Diese Betriebsart nennt man »One Shot Mode«. Andernfalls beginnt der Timer erneut von dem Wert, der vorher eingegeben wurde, nach Null zu zählen (»Continuous Mode«). Starten und stoppen kann man den Timer jederzeit mit Bit 0 des jeweiligen Kontrollregisters. Setzt man das Bit, so startet der Timer, löscht man es, bleibt er stehen. Eine Übersicht über alle Betriebsarten gibt Tabelle 1. Die Timer können in konstanten Zeitabständen (bei jedem Unterlauf) einen Interrupt erzeugen, wenn ein bestimmtes Bit im Interrupt-Control Register (Register 13, siehe auch Tabelle 1) gesetzt wird. Dieser Interrupt-Impuls, der von Timer A der CIA 1, im Normalfall jede 1/60-Sekunde ausgelöst wird, bewirkt einen Sprung ins Interrupt-Programm, mit dem unter anderem die Tastatur abgefragt und die Software-Uhr (TI$ und TI) gestellt wird. Ein interessanter Effekt entsteht, wenn man den Wert, von dem dieser Timer nach Null zählt, verändert. Normalerweise ist dieser Wert 16427 (High-Byte = 64, Low-Byte = 37). Verkleinert man diesen Wert, indem man zum Beispiel »5« in das High-Byte-Register schreibt (POKE 56325,5), dann beginnt der Cursor äußerst schnell zu blinken. Dies liegt daran, daß jetzt die Interrupts in kürzerer Folge ausgelöst werden. Auch die Software-Uhr TI$ läuft nun viel schneller. Vergrößert man den Wert (etwa durch POKE 56325,200) geschieht genau das Umgekehrte. Der Cursor blinkt ermüdend langsam und die »TI$-Uhr« geht nach.

Über die eben erwähnten»Uhren« (TI,TI$) werden sich diejenigen, denen es auf hohe Genauigkeit ankommt, schon geärgert haben. Sie haben nämlich eine Ungenauigkeit von bis zu einer halben Stunde pro Tag. Außerdem werden sie bei Kassettenoperationen ganz abgeschaltet. In der CIA ist jedoch eine Uhr mit hervorragender Genauigkeit und einer Auflösung von einer Zehntelsekunde.

Sehr ganggenaue Echtzeituhr

Diese Uhr wird durch die Netzfrequenz gesteuert (50 Hz), wodurch auch die hohe Präzision zu erklären ist. Man kann sogar eine Alarmzeit vorwählen, bei der die CIA einen Interrupt auslöst. Die Uhr belegt in der CIA die Register 8 bis 11 (Tabelle 1). Die Zeitwerte stehen in den Registern im BCD-Format. BCD ist die Abkürzung für »Binary Code Decimal«. Das heißt, daß jeweils 4 Bits (1 Nibble) einer Binärzahl zu einer Dezimalstelle zusammengefaßt werden. Dabei darf natürlich der Wert eines Nibbles nicht größer als 9 sein. Bei einer 8-Bit-BCD-Zahl ergibt sich also ein Höchstwert von dez. 99 (=bin 1001 1001) im Gegensatz zu dez. 255 bei einer reinen Binärzahl. Die Kombination 0001 0011 ergibt dezimal 13, denn 0001 ergibt 1 und 0011 ergibt 3. Bei Register 11 (Stunden) ist noch zu beachten, daß Bit 7, ähnlich wie bei einer Digitaluhr, das AM/PM-Flag (Vormittag/Nachmittag) darstellt. Es ist Nachmittag, wenn dieses Bit gesetzt ist. Um eine echte 24-Stunden-Anzeige zu bekommen, müssen zu dem Stundenregister 12 Stunden hinzugezählt werden, wenn Bit 7 gesetzt ist. Sehen wir uns nun ein Beispiel für die Programmierung der Uhr an.

Nehmen wir an, die Uhrzeit soll auf 15:45:32,5 h gestellt werden. Der erste Schritt besteht darin, Bit 7 in Register 14 zu setzen (POKE 56589, PEEK(56589) OR 128). Hiermit wird die Uhr auf die Verarbeitung einer Netzfrequenz von 50 Hz eingestellt, anstelle der in USA üblichen 60 Hz. Dieses Flag braucht nur einmal auf 50 Hz gesetzt zu werden. Bei einem Reset durchläuft der Computer eine Routine, die das Flag wieder auf 60 Hz umschaltet. Der zweite Schritt besteht darin, der CIA mitzuteilen, daß die Uhrzeit und nicht die Alarmzeit gesetzt werden soll. Dazu löscht man Bit 7 von Register 15 (POKE 56590, PEEK(56590) AND 127). Dieses Bit wird beim Einschalten und nach einem Reset auf »0« gesetzt (Uhrzeit setzen). Jetzt kann man beginnen, die Uhrzeit in die entsprechenden Register zu schreiben. Hierbei muß mit den Stunden begonnen werden, weil durch einen Schreibzugriff auf das Stundenregister die Uhr anhält. Dies hat durchaus einen Sinn, denn wer kommt schon auf die Idee, eine weiterlaufende Digitaluhr zu stellen? Das Register wird auf 15 Uhr gestellt, indem man die BCD-Zahl für 3 Uhr in das Register schreibt und zusätzlich Bit 7 auf Nachmittag setzt (PM). Es muß eingegeben werden: POKE 56587, 3:POKE 56587, PEEK(56587) OR 128 (oder POKE 56587, 131). Da die Uhr jetzt angehalten ist, können in aller Ruhe die Minuten, Sekunden und Zehntelsekunden gesetzt werden. Zur Einstellung der Minuten ist anzugeben: POKE 56586, 39 (45 Minuten) für die Sekunden: POKE 56585, 50 (32 Sekunden). Für die Zehntelsekunden: POKE 56584, 5. Die Zehntelsekunden müssen als letzte angegeben werden, da durch einen Schreibzugriff auf dieses Register die Uhr erneut gestartet wird. Es ist also unbedingt nötig, nach einem Stellen der Uhr, auch wenn es einem nicht auf eine Zehntelsekunde ankommt, dieses Register zu beschreiben. Beim Setzen der Alarmzeit wird analog verfahren. Es muß jedoch Bit 7 von Register 15 gesetzt werden (POKE 56591, PEEK (56591) OR 128). Die Uhr wird natürlich beim Stellen der Alarmzeit nicht angehalten. Bei einer Übereinstimmung von Uhrzeit und Alarmzeit wird Bit 2 gesetzt und gegebenfalls ein Interrupt ausgelöst. Ein Beispielprogramm für den Umgang mit der Echtzeituhr zeigt Listing 1.

Serielle Datenübertragung: Bit für Bit im Gänseschritt

Neben der parallelen Datenübertragung, die wir bereits im Abschnitt über den Daten-Port kennengelernt haben, gibt es noch eine andere Übertragungsart, bei der die einzelnen Bits eines Bytes nicht »nebeneinander«, sondern »hintereinander« übermittelt werden. Diese Übertragungsart heißt seriell. Die Methode hat den Vorteil, daß man weniger Datenleitungen braucht, prinzipiell nämlich nur noch eine anstelle von in der Regel acht beim parallelen Datenverkehr. Der Nachteil ist aber

Reg.	Adresse		Funktion	Verwendung
0	56320 56576	($dc00) ($dd00)	Datenregister Port A Ein gesetztes Bit signalisiert High an der entsprechenden Port-Leitung	Tastaturabfrage IEC-Bus + RS232
1	56321 56577	($dc01) ($dd01)	Datenregister Port B Wie Register 0, jedoch für Port B	Tastaturabfrage User-Port
2	56322 56578	($dc02) ($dd02)	Datenrichtungsregister Port A Ein gesetztes Bit programmiert die zugehörige Portleitung als Ausgang	zusammen mit Register 0 zusammen mit Register 0
3	56323 56579	($dc03) ($dd03)	Datenrichtungsregister Port B Wie Register, jedoch für Port B	zusammen mit Register 1 zusammen mit Register 1
4	56324 56580	($dc04) ($dd04)	Timer A, Low-Byte Beim Lesen wird der momentane Zählerstand erhalten, beim Schreiben der Zählerstand (Low-Byte) gesetzt, von dem der 16-Bit-Zähler nach Null zählt	IRQ (alle 1/60 s) RS232
5	56325 56581	($dc05) ($dd05)	Timer A, High Byte Wie Register 4, jedoch für High-Byte, Timer A Siehe auch Register 14 (Control-Register A)	zusammen mit Register 4 zusammen mit Register 4
6	56326 56582	($dc06) ($dd06)	Timer B, Low-Byte Wie Register 4, jedoch für Timer B Siehe auch Register 15 (Control-Register B)	für Kassetten Op. RS232
7	56327 56583	($dc07) ($dd07)	Timer B, High-Byte Wie Register 5, jedoch für Timer B Siehe auch Register 15 (Control-Register B)	zusammen mit Register 6 zusammen mit Register 6
8	56328 56584	($dc08) ($dd08)	Time of Day 1/10 Sekunden Bit 0-3 enthalten die 1/10 Sekunden im BCD-Format. Ist Bit 7 in Register 15 gesetzt, so wird beim Schreiben die Alarmzeit gesetzt, ansonsten die Uhrzeit. Bit 4-7 unbenutzt.	(für RND) unbenutzt
9	56329 56585	($dc09) ($dd09)	Time of Day Sekunden Dieses Register enthält die Sekunden im BCD-Format. Schreibzugriff siehe Register 8	(für RND) unbenutzt
10	56330 56586	($dc0a) ($dd0a)	Time of Day Minuten Dieses Register enthält die Minuten im BCD-Format Schreibzugriff siehe Register 8	(für RND) unbenutzt
11	56331 56587	($dc0b) ($dd0b)	Time of Day Stunden Bit 0-3 enthalten die Stunden im BCD-Format, Bit 4 die 10er Stunden. Bit 7 ist bei AM (vormittags) 0 und bei PM (nachmittags) 1. Bit 5+6 unbenutzt Schreibzugriff siehe Register 8	(für RND) unbenutzt
12	56332 56588	($dc0c) ($dd0c)	Serial Data Register (SDR) Schieberegister, über das Daten am Pin SP herausgeschoben und hereingeholt werden. Das höchstwertige Bit erscheint zuerst.	unbenutzt unbenutzt
13	56333 56589	($dc0d) ($dd0d)	Interrupt Control Register (ICR) Bit 0: Unterlauf Timer A Bit 1: Unterlauf Timer B Bit 2: Uhrzeit und Alarmzeit sind gleich Bit 3: Schieberegister voll oder leer (je nach Betriebsart) Bit 4: 1, wenn negative Spannungsflanke an FLAG aufgetreten ist Bit 5 und Bit 6 sind immer 0 Bild 7: Es stimmt mindestens ein gesetztes Bit im INT MASK und INT DATA-Register überein Achtung: Beim Lesen wird das ICR gelöscht!	
14	56334 56590	($dc0e) ($dd0e)	Control Register A (CRA) Bit 0: 1=Timer A starten 0=Timer A stoppen Bit 1: 1=Ein Umlauf von Timer A wird an PB 6 signalisiert, auch wenn dieses Port-Bit als Eingang programmiert ist. Bit 2: 1=Bei einem Unterlauf von Timer A wird PB 6 invertiert Bit 3: 0=Continous-Mode 1=One-Shot-Mode Bit 4: Wird eine 1 eingeschrieben, so wird Timer A sofort mit dem Wert geladen, der vorher in Register 4 + 5 stand, egal ob der Timer gerade läuft oder nicht. Bit 5: 1=Timer A zählt positive Flanken an CNT 0=Timer A zählt Systemtakte Bit 6: 0=Das Schieberegister ist Eingang 1=Das Schieberegister ist Ausgang Bit 7: 1=TOD verarbeitet 50 Hz Netzfrequenz 0=TOD verarbeitet 60 Hz Netzfrequenz	
15	56335 56591	($dc0f) ($dd07)	Control Register B (CRB) Bit 0-4: entsprechen Bit 0-4 von CRA, jedoch für Timer B und PB 7 Bit 5+6 bestimmen paarweise die Triggerquelle 00=Timer B zählt Systemtakte 01=Timer B zählt positive Flanken an CNT 10=Timer B zählt Unterläufe von Timer A 11=Timer B zählt Unterläufe von Timer A nur, wenn CNT high ist Bit 7: 1=TOD Alarmzeit setzen 0=TOD Uhrzeit setzen	

Tabelle 1. Die Register der CIA

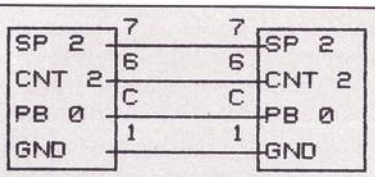

Bild 2. So können Sie zwei C 64 über die User-Ports koppeln. Verbinden Sie die angegebenen Anschlüsse.

die längere Zeit, die man zur Übertragung der Daten benötigt. Der Teil der CIA 6526, der für die serielle Datenübertragung zuständig ist, ist das serielle Datenregister (SDR, Register 12). Ob der Chip die Daten selbst senden oder empfangen soll, bestimmt Bit 6 des Kontrollregisters A (Registers 14). Ist das Bit im Kontrollregister gesetzt, arbeitet das SDR als Ausgang, ansonsten als Eingang. Wenn das Register als Ausgang arbeitet, wird sofort, nachdem ein beliebiger Wert in Register 12 (Serielles Datenregister) geschrieben wurde, damit begonnen, die 8 Bits dieses Wertes nacheinander über den Pin 39 der CIA auszugeben. Dieser Pin ist für CIA 1 am User-Port über Anschluß 5 erreichbar, für CIA 2 an Anschluß 7. Der Timer A der CIA wird dazu benutzt, die Geschwindigkeit der Ausgabe festzulegen. Bei jedem zweiten Unterlauf des Timers (wenn er beim Herunterzählen eines Wertes in den negativen Zahlenbereich gelangt) wird ein Bit des Wertes, der im seriellen Datenregister steht, ausgegeben, wobei das höchstwertigste Bit (Bit 7) dieses Wertes als erstes erscheint. Hierbei ist es natürlich notwendig, den Timer auf Continuous-Mode einzustellen, was durch Löschen des Bit 3 von Register 14 erreicht wird. Jedesmal, wenn ein Bit ausgegeben wurde, erscheint am Pin CNT ein kurzer Low-Impuls (also ein Spannungswechsel von +5 Volt auf 0 Volt und zurück), der einem möglichen Empfänger signalisiert, daß ein »neues« Bit zur Übergabe bereitsteht. Soll der Computer Daten empfangen, so muß Bit 6 vom Kontrollregister A (Register 14) gelöscht werden. Das SDR arbei-

tet dann als Eingang. Ein Bit wird immer dann in das Schieberegister übertragen, wenn am Pin CNT des empfangenden ICs ein Low-Impuls erscheint. Das höchstwertigste Bit wird hier, wie auch beim Senden, zuerst in das serielle Datenregister übernommen. Bit 3 des Interrupt-Control-Registers (Register 13) wird gesetzt, wenn das Schieberegister vollständig gefüllt oder geleert worden ist. Mit den Schieberegistern, die im C 64 übrigens völlig unbenutzt sind, können wir zum Beispiel zwei Computer miteinander Daten austauschen lassen. Dazu sind in Listing 2 und Listing 3 zwei kurze Basic-Programme abgedruckt. Die beiden C 64 müssen nur mit einem vierpoligen Kabel — wie in Bild 2 gezeigt — verbunden werden. Der C 64, der Daten senden soll, muß das Sendeprogramm (Listing 2) im Speicher haben, der Empfänger natürlich das Empfangsprogramm (Listing 3). Das Sendeprogramm arbeitet folgendermaßen: In Zeile 110 wird der Variablen CIA die Basisadresse der zweiten CIA zugeordnet. Dann wird die Timer-Rate festgelegt (Zeile 120 und 130), zuerst das Low- und dann das Highbyte. Hier wurde die schnellste Timerrate gewählt ($0002). In den folgenden drei Zeilen werden nacheinander von Register 14 (Kontrollregister A) Bit 3 gelöscht (Continuous-Mode), Bit 0 (Timer A startet) und Bit 6 gesetzt (Schieberegister arbeitet als Ausgang). In Zeile 170 wird der Daten-Port B als Eingang geschaltet, denn Bit 0 dieses Ports dient in unserem Programm als »Habe die Daten empfangen«-Leitung, über die der Empfänger dem Sender mitteilt, daß die Daten angekommen sind.

Datenübertragung über den User-Port

Danach werden in Zeile 180 und 190 die Startadresse, ab der die zu übertragenden Daten im Speicher stehen, und die Anzahl der Daten eingegeben. Jetzt kommt die Schleife, in der die Daten übertragen werden. In Zeile 210 wartet der Sender auf das Bereitschaftssignal des Empfängers. Dann wird der erste Wert aus dem Speicher geholt und in das Schieberegister gebracht. Um die eigentliche Übertragung kümmert sich dann die CIA. Nachdem der Empfänger wieder ein Frei-Zeichen gegeben hat, wird der nächste Wert in das Schieberegister geschrieben und das Spiel beginnt von neuem. Das Empfangsprogramm legt in Zeile 110 zunächst, wie auch das Sendeprogramm, die Basisadresse der zweiten CIA in der Memory Map des C 64 fest. Danach wird das Schieberegister auf Eingang geschaltet (Zeile 120) und Port B auf Ausgang (Zeile 130). Nun wird in Zeile 140 und 150 der Speicherbereich festgelegt, in dem die Daten gespeichert werden. Bei »Anzahl« muß natürlich die gleiche Zahl eingegeben werden wie beim Sender. In der folgenden Empfangsschleife wird als erstes ein Impuls auf der Bereitschaftsleitung erzeugt (Zeile 170-190). Die Schleife in Zeile 180 ist eine Verzögerungsschleife. Das Programm wartet in Zeile 200, bis alle 8 Bits eingetroffen sind, und speichert schließlich den empfangenen Wert ab (Zeile 210). Wenn beide Computer über den User-Port mit dem Kabel verbunden sind und sich die beiden Programme im Speicher befinden, kann es losgehen. Zum Test könnte man den Inhalt des Bildschirms übertragen, indem bei beiden Rechnern als Startadresse 1024 eingegeben wird und als Anzahl 1000. Es ist hierbei zu beachten, daß die Eingaben zuerst beim Sendeprogramm abgeschlossen werden, denn sonst kann es passieren, daß der Sender das Bereitschaftssignal des Empfängers nicht mitbekommt und dann in der Schleife (Zeile 210) steckenbleibt. Wenn auf dem Bildschirm des Empfängers nichts oder nur Bruchstücke zu lesen sind, liegt das daran, daß die Zeichen in der Hintergrundfarbe erscheinen. Man muß also in das Farb-RAM auch noch einen anderen Farbcode schrei-

```
110 REM *    BEISPIELPROGRAMM ZUR     *      <077>
120 REM *   NUTZUNG DER ECHTZEITUHR    *      <087>
140 CIA = 56576:REM BASISADRESSE CIA #2       <012>
150 REM 50 HERTZ NETZFREQUENZ                 <077>
160 POKE CIA+15,PEEK(CIA+15) OR 128           <028>
170 PRINT "ALARMZEIT SETZEN"                  <239>
180 GOSUB 1000                                <136>
190 GOSUB 2000                                <154>
200 REM ALARMZEITFLAG                         <193>
210 POKE CIA + 15,PEEK (CIA+15) OR 128        <078>
220 GOSUB 3000                                <192>
230 PRINT "UHRZEIT SETZEN"                    <230>
240 GOSUB 1000                                <196>
250 GOSUB 2000                                <214>
260 REM UHRZEITFLAG                           <092>
270 POKE CIA + 15,PEEK (CIA+15) AND 127       <067>
280 GOSUB 3000                                <254>
290 GOSUB 4000                                <016>
300 PRINT CHR$ (147)                          <075>
310 PRINT H;":";M;":";S;",";Z                 <228>
320 REM ALARM ?                               <239>
330 A = PEEK (CIA+13)                         <016>
340 POKE CIA+13,A                             <092>
350 IF (A AND 4) = 0 GOTO 290                 <219>
360 FOR I=0 TO 20:PRINT "ALARM":NEXT          <235>
370 END                                       <118>
1000 REM EINGABE                              <240>
1010 INPUT"0 = AM{2SPACE}1 = PM ";FL          <085>
1020 IF FL <> 0 AND FL <> 1 THEN 1010         <061>
1030 INPUT"STD.,MIN.,SEK.,1/10 SEK";H,M,S,
     Z                                        <197>
1040 RETURN                                   <082>
1050 RETURN                                   <092>
2000 REM UMRECHNUNG IN BCD-FORMAT             <008>
2010 H1=INT (H/10)                            <185>
2020 H2 = H - H1*10                           <181>
2030 H = H1*16 + H2                           <092>
2040 M1 = INT (M/10)                          <109>
2050 M2 = M - M1*10                           <011>
2060 M = M1*16 + M2                           <244>
2070 S1 = INT (S/10)                          <168>
2080 S2 = S - S1*10                           <239>
2090 S = S1*16 + S2                           <163>
2100 RETURN                                   <126>
3000 REM REGISTER SETZEN                      <250>
3010 H = H + FL * 128:REM AM/PM               <067>
3020 POKE CIA + 11,H                          <002>
3030 POKE CIA + 10,M                          <031>
3040 POKE CIA + 9,S                           <066>
3050 POKE CIA + 8,Z                           <218>
3060 RETURN                                   <068>
4000 REM REGISTER LESEN                       <132>
4010 H = PEEK (CIA+11)                        <152>
4020 FL = (H AND 128)/ 128 * 12               <162>
4030 H1 = (H AND 16) / 16 * 10                <025>
4040 H = H1 + FL + (H AND 15)                 <151>
4050 M = PEEK (CIA+10)                        <204>
4060 M1 = (M AND 112) / 16 * 10               <042>
4070 M = M1 + (M AND 15)                      <049>
4080 S = PEEK (CIA+9)                         <088>
4090 S1 = (S AND 112) / 16 * 10               <225>
4100 S = S1 + (S AND 15)                      <208>
4110 Z = PEEK (CIA+8) AND 15                  <253>
4120 RETURN                                   <114>
```

Listing 1. Beispielprogramm zur Echtzeituhr

```
100 REM *** SENDEN ***                        <030>
110 CIA=56576                                 <018>
120 POKE CIA+4,2                              <000>
130 POKE CIA+5,0                              <134>
140 POKE CIA+14,PEEK(CIA+14) AND 247          <012>
150 POKE CIA+14,PEEK(CIA+14) OR 1             <187>
160 POKE CIA+14,PEEK(CIA+14) OR 64            <155>
170 POKE CIA+3,0                              <173>
180 INPUT "STARTADRESSE";SA                   <175>
190 INPUT "ANZAHL";AZ                         <152>
200 FOR I = SA TO SA+AZ                       <125>
210 : IF (PEEK(CIA+1) AND 1) = 0 THEN 210     <090>
220 : POKE CIA+12,PEEK(I)                     <210>
230 NEXT I                                    <058>
240 END                          © 64'er      <242>
```

Listing 2. Sendeprogramm zur Datenübertragung

ben. Die beiden Programme stellen jedoch nur eine Anregung zum Experimentieren mit den Schieberegistern dar. Es wäre zum Beispiel möglich, durch Verwendung von Maschinensprache die Übertragungsgeschwindigkeit zu erhöhen, etc.

Statusanzeige der CIA

Am Inhalt des Interrupt Control Registers kann man bestimmte Ereignisse oder Zustände der CIA feststellen. Durch Schreiben in dieses Register kann man aber auch festlegen, welches Ereignis zusätzlich einen Interrupt auslösen soll. Jedem Bit dieses Registers ist ein Ereignis zugeordnet.
Bit 0: Unterlauf von Timer A
Bit 1: Unterlauf von Timer B
Bit 2: Echtzeituhr, Alarmzeit erreicht
Bit 3: Schieberegister ist voll/leer, je nach Betriebsart
Bit 4: Low-Pegel am Pin FLAG aufgetreten.
Bit 5 und 6 sind unbenutzt. Beim Interrupt Control Register (Register 13) muß man zwischen Schreib- und Lesezugriff unterscheiden. Schreibzugriff: Hierbei kann man die Interruptquelle(n) bestimmen. Ist Bit 7 des Interrupt Control Register gesetzt, lösen die Ereignisse, die über die Bits 0 bis 4 festgelegt wurden, einen Interrupt aus. Anders ist es, wenn Bit 7 gelöscht ist. Dann werden nämlich alle Interrupts gesperrt, deren Bits gesetzt sind. Die Interrupts, deren Bits nicht gesetzt werden, bleiben in ihrem vorhergehenden Status. Soll zum Beispiel die Echtzeituhr beim Erreichen der Alarmzeit einen Interrupt auslösen, kann man folgende Befehle eingeben: POKE (Register 13), 31 (=bin. 00011111), wodurch sämtliche Interrupts zunächst gesperrt werden, weil es sein kann, daß vorher schon der eine oder andere Interrupt freigegeben wurde. Danach kann man den Alarmzeit-Interrupt durch POKE (Register 1), 132 (=bin. 10000100) freigeben. Lesezugriff: Beim Lesen des Interrupt Control Register erfährt man nun, welches Ereignis aufgetreten ist. Unabhängig davon, ob es als Interrupt-Anforderung zugelassen wurde. Ist ein Bit gesetzt, ist das zugehörige Ereignis aufgetreten. Zusätzlich ist Bit 7 gesetzt, wenn der Interrupt freigegeben war und ausgelöst wurde. Beim Umgang mit diesem Register ist jedoch Vorsicht geboten, da es beim Lesen gelöscht wird!

Der Hauptunterschied der beiden CIAs im C 64 untereinander ist, daß CIA 1 (Basisadresse 56320) ausschließlich IRQs auslösen kann, CIA 2 (56576) nur NMIs. NMI und IRQ sind verschiedene Interrupt-Arten. Ein IRQ (Interrupt-Request) ist der Interrupt, der normalerweise alle 1/60-Sekunde vom Timer A der CIA 1 ausgelöst wird. Während des Interrupts wird beispielsweise die Tastatur abgefragt. Daher sollte man diesen Interrupt im Direktmodus nicht sperren. Der Computer würde das sofort mit einem Absturz quittieren. Ein NMI (Non-Maskable-Interrupt) wird beispielsweise auch durch die RESTORE-Taste ausgelöst, was aber nichts mit der CIA zu tun hat.

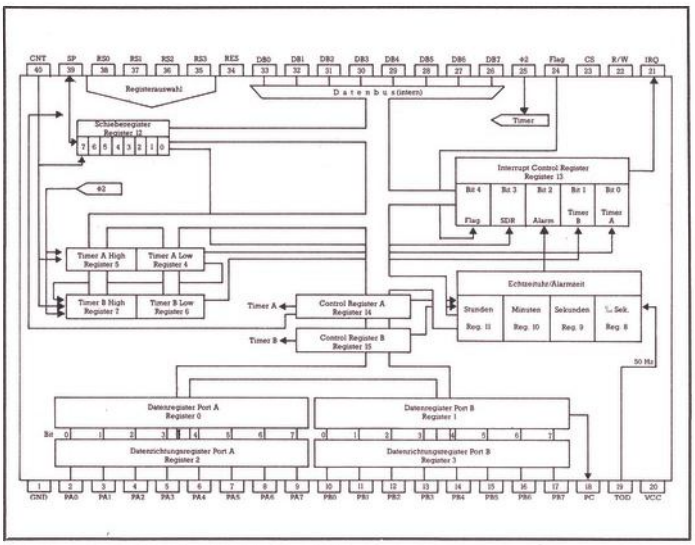

Bild 1. Blockschaltbild der CIAs

```
100 REM *** EMPFANGEN ***                          <102>
110 CIA=56576                                      <018>
120 POKE CIA+14,PEEK(CIA+14) AND 191               <008>
130 POKE CIA+3,255                                 <006>
140 INPUT "STARTADRESSE";SA                        <135>
150 INPUT "ANZAHL";AZ                              <112>
160 FOR I = SA TO SA+AZ                            <085>
170 : POKE CIA+1,1                                 <157>
180 : FOR T = 0 TO 1 : NEXT T                      <242>
190 : POKE CIA+1,0                                 <173>
200 : IF (PEEK(CIA+13) AND 8) = 0 THEN 200         <174>
210 : POKE I,PEEK(CIA+12)                          <037>
220 NEXT I                                         <048>
230 END                        © 64'er             <232>
```

Listing 3. Empfangsprogramm zur Datenübertragung

Die CIA und ihre Aufgaben

Die beiden Ports der CIA 1 (IRQ-CIA) werden zur Abfrage der Tastatur und von eventuell angeschlossenen Joysticks verwendet. Die Tastatur ist als eine 8x8-Matrix geschaltet. Über Port A werden ständig nacheinander die acht Matrixzeilen abgetastet. Über Port B kommt dann die Rückmeldung, welche der Tasten in der Zeile gedrückt war. Ist die Tastatur abgeschaltet und stattdessen Joysticks in den Control-Ports eingesteckt, ist Port A für den Control-Port 2 zuständig, Port B zeigt den Zustand von Control-Port 1. Timer A gibt dem Prozessor den Abfrage (Interrupt)-Takt vor. Bei jedem Interrupt wird zum Beispiel die Tastatur nach der eben beschriebenen Methode abgefragt und die Software-Uhr TI und TI$ gestellt. Außerdem wird Timer A, zusammen mit Timer B, für Kassetten-Operationen gebraucht. Timer B regelt, neben Kassetten-Operationen, auch Zugriffe auf den seriellen Bus.

Die Echtzeituhr der CIA 1 wird vom Betriebssystem nur dazu benutzt, Zufallszahlen (RND(0)) zu erzeugen. Dies beeinflußt aber andere Programme nicht. Der serielle Port ist unbenutzt. Er steht dem Anwender am User-Port, Anschluß 5, zur Verfügung.

Die zweite CIA (NMI-CIA) hat weniger Aufgaben als die CIA 1. Bit 0 und 1 des Ports A dienen zur Auswahl der Video-Speicher-Bank. Bit 2 hat nur im Zusammenhang mit einer RS232-Schnittstelle eine Aufgabe. Bit 3 bis 7 regeln den Datenverkehr auf dem seriellen Bus für Drucker und Diskettenlaufwerke. Die Bedeutung ist dabei folgende:
Bit 3: ATN out (Attention out)
Bit 4: CLK out (Clock out, Taktausgang)
Bit 5: SER out (Serial out, Datenausgang)
Bit 6: CLK in (Clock in, Takteingang)
Bit 7: SER in (Serial in, Dateneingang)

Der Port B kann vollständig am User-Port benutzt werden. Er hat sonst keine Funktion. Die Anschlüsse sind C bis L. Die übrigen Teile der CIA 2 sind im C 64 unbenutzt und stehen dem Anwender ebenfalls zur freien Verfügung. Eine Ausnahme bilden hier die Timer A und B, die den Datenverkehr auf der RS232-Schnittstelle regeln. Probieren Sie einfach einmal aus, was Sie alles mit den CIAs anfangen können. Denn wie schon gesagt, es gibt einige Funktionen der CIAs, die der C 64 nicht nutzt. (Bernhard Binz/Christian Mück/hm)