

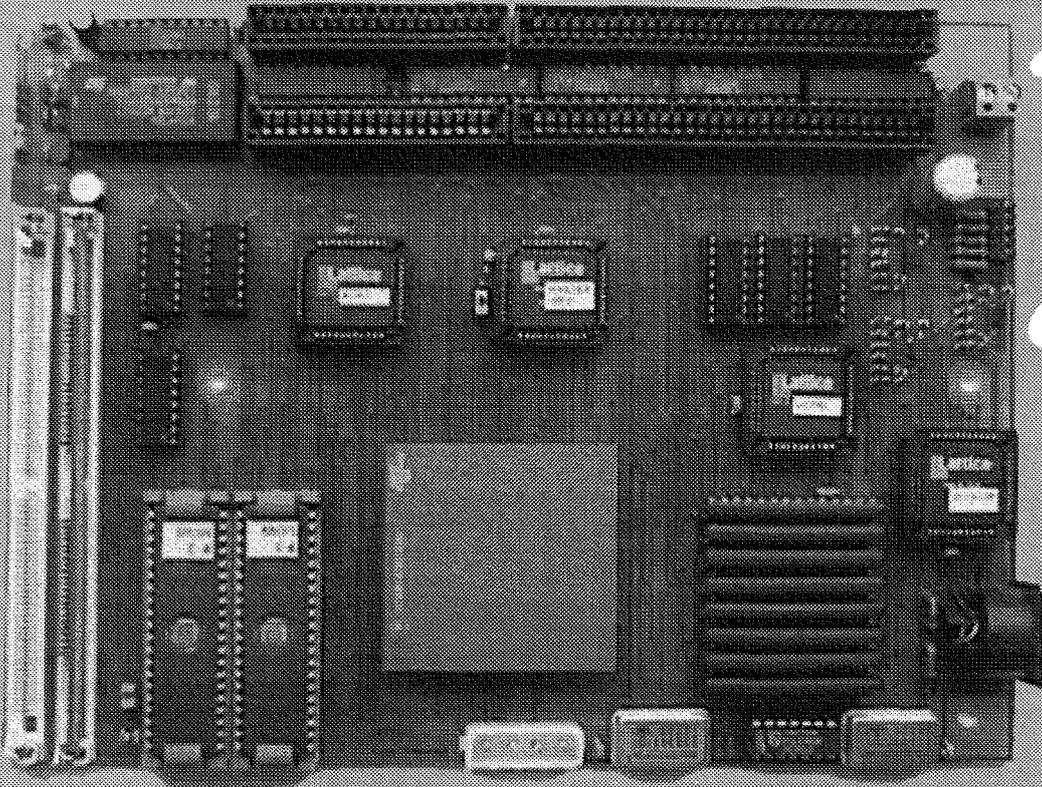
QL Today

DEUTSCH

Jahrgang 2
Ausgabe 5
Januar/Februar
1998

ISSN 1432-5446

Das Magazin über QL, QDOS,
Sinclair Computer, SMSQ...



**Q40 - der
QL-Nachfolger in
greifbarer Nähe!**

Herausgeber:

Jochen Merz Software Tel. +49 203 502011
Im stillen Winkel 12 Fax +49 203 502012
47169 Duisburg Box1 +49 203 502013
Deutschland Box2 +49 203 502014
email: JMerz@j-m-s.com

QL Today erscheint alle zwei Monate, Erscheinungsdatum der ersten Ausgabe ist der 15. Mai. Das Abo beginnt mit der aktuellen Ausgabe zum Zeitpunkt der Bestellung.

Informationen über Abonnenten-Preise und Antragsformulare sind bei Jochen Merz Software erhältlich.

Bezahlung kann in DM erfolgen, entweder mit Verrechnungsscheck einer Bank mit Sitz in Deutschland oder Euroscheck. Schecks sollten auf Jochen Merz Software ausgestellt sein. Es besteht auch die bequeme Möglichkeit der Einzugsermächtigung, auch hier nur bei Banken in Deutschland. Zahlung per Kreditkarte ist ebenfalls möglich - hier wird neben Ihrer Kartenummer auch die Gültigkeitsdauer benötigt.

Ihre Kommentare, Vorschläge und Artikel sind herzlich willkommen. SIE machen **QL Today** möglich. Wir verbessern das Magazin wo immer möglich, um Ihren Vorstellungen gerecht zu werden. Artikel sollten auf 3,5" Diskette (DD oder HD) eingeschickt werden. Das Format sollte ASCII, Quill oder Text87 (Druckertreiber angeben!) sein. Bilder sollten im _SCR-Format geschickt werden, GIF und TIF ist auch möglich. BITTE senden Sie auch einen Ausdruck der Bilder. Wenn ein Bild an einer bestimmten Stelle plaziert werden soll, geben Sie es bitte auch an.

Redaktionsschluß für Artikel und Werbung:

Ausgabe 1:	15. April
Ausgabe 2:	15. Juni
Ausgabe 3:	15. August
Ausgabe 4:	15. Oktober
Ausgabe 5:	15. Dezember
Ausgabe 6:	15. Februar

QL Today behält sich vor, eingeschicktes Material nicht zu veröffentlichen. **QL Today** ist unter keinen Umständen für die Richtigkeit der abgedruckten Artikel und Programmen haftbar, ebenso nicht für aus fehlerhaftem Material hervorgerufene Datenverluste, Unbenutzbarkeit oder ähnliche Probleme, die aus Artikeln in **QL Today** herrühren könnten. Die Meinung in diesem Magazin entspricht der des jeweiligen Autors und nicht notwendigerweise der des Herausgebers.

Inhalt

- 3 Editorial
- 4 Grafikkarte & Monitor - Rolf Ritter
- 5 Das QXL.WIN-Format - Teil 1 - Gerhard Plavec
- 7 Jochen Merz Software
- 10 **Titelstory:** Q40 - Projekt eines modernen QL - Peter Graf
- 11 Software-Updates über's WWW - Jochen Merz
- 12 Uhren-Korrektur - Dietrich Buder
- 14 Versuch eines Berichts vom 6. Internationalen Spectrum-User-Club Treffen in Sindelfingen am 15.11.1997 - Otmar Peters
- 16 Zusammenfassung der englischen Ausgabe 4 - Jochen Merz
- 18 QL-Treffen Österreich

Kleinanzeigen

Da QL Today eine der wichtigsten Quellen für QL-Neuigkeiten ist, werben auch die meisten QL-Händler hierin. Nun stellt sich natürlich die Frage, warum nur QL-Händler inserieren dürfen, warum nicht auch QL-User, die ihre eigenen Programme, Hardware oder Entwicklungen verkaufen möchten.

Anfangs konnte man selbstgeschriebene Software von Clubs vertreiben lassen, beispielsweise dem deutschen QL User Club. Aber warum soll man es denn nicht selbst vermarkten können? Es gibt hier keinen Unterschied zwischen privaten und kommerziellen Anzeigen, nur mit dem QL und Drumherum sollte es schon etwas zu tun haben.

Bis zu 50 Worte im englischen oder deutschen Teil kosten DM 5,- (oder 3 Internationale Antwortscheine), bis zu 100 Worte kosten DM 10,- (oder 6 Internationale Antwortscheine). Soll die Anzeige im deutschen und englischen Teil erscheinen, ist der Preis zu verdoppeln.

Dieses Magazin unterliegt dem Copyright und jegliches hierin veröffentlichte Material darf nicht ohne schriftliche Erlaubnis von **QL Today** reproduziert, übersetzt oder sonstwie verbreitet werden. Allen Copyrights und Trademarks wird hiermit Rechnung getragen.

Editorial

Jochen Merz

Liebe Leser,

ich hoffe, Ihr hattet einen guten Rutsch ins neue Jahr. Ich werde nun keinen Jahresrückblick präsentieren sondern statt dessen auflisten, was es denn so alles Neues gibt.

Fangen wir mit Eindhoven an: Tony Tebby kam und stellte seinen lang ersehnten neuen Bildschirmtreiber vor. Die mitgebrachte Version stürzte aus mir unbekanntem Gründen auf einem QXL-Testrechner ab, aber Tony hatte noch einen Stapel Papier mit unglaublich interessantem Inhalt dabei: sein Bildschirmtreiber-Design. Ich möchte jetzt noch nicht auf Details eingehen, aber soviel kann ich jetzt schon sagen: Wenn alles so wird wie geplant, bekommen wir etwas, das alles Gute von anderen Systemen kann und trotzdem noch besser funktioniert (und auch gut aussieht!).

Ein weiteres Highlight des Treffens war die funktionierende Vorstellung der Q40-Karte. Im Sommer in Solms war sie ja schon zu besichtigen, doch wenn etwas dann tatsächlich lauffähig demonstriert wird, ist es wesentlich beeindruckender. Peter Graf, Designer der Q40, hat dazu auch etwas in dieser Ausgabe geschrieben. Alle waren von dem Board begeistert und es hat sicherlich alle Möglichkeiten, der nächste (und damit der erste richtige) "QL-Nachfolger" zu werden.

Die ROM-Disk: Viele warten drauf, hoffentlich ist sie zu dem Zeitpunkt, zu dem Sie dies hier lesen auch erhältlich. Die Hardware steht schon und die Vorbestellungen liegen auch schon vor, doch ist die Entwicklung der Treiber-Software deutlich schwieriger als anfangs gedacht. Hauptproblem ist die Tatsache, daß die Sektoren verschoben werden müssen. Tony hat beim Design der Treiber viel Wert darauf gelegt, daß möglichst wenig Änderungen pro Schreibzugriff erfolgen, da jeder 64k-Sektor der ROM-Disk "nur" ca. 100.000 mal löschbar ist. Damit eine möglichst gleichmäßige Benutzung gewährleistet ist, müssen die Sektoren verschoben werden. Dies war bei keinem existierenden Treiber (RAM, FLP, MDV, WIN) notwendig, da hier ein zugewiesener Sektor immer an der gleichen Stelle bleibt. Es muß nun viel mehr umgeschrieben werden als gedacht, so auch die Behandlung der Slave-Blöcke und dergleichen. Wenn man bedenkt, was beim Schreiben eines Bytes in eine Datei alles passiert, dann ist es ersichtlich, daß ein vernünftiges Design notwendig ist: Ein neuer Sektor könnte benötigt werden, die Dateilänge im Dateiheder ändert sich, die Map (Karte der vergebenen Sektoren) und das Ver-

zeichnis ändern sich. Hinzu kommt, daß immer nur 64k-Blöcke hardwaremäßig gelöscht werden können und Bits von 1 auf 0 (wie beim EPROM) geändert werden. Aber wie gesagt, Tony hat für all diese Probleme intelligente Lösungen parat.

Ich möchte allen Autoren dieser Ausgabe für Ihre Mitarbeit danken, ebenso Dietrich Buder wieder für's konstruktive Korrektur-Lesen und würde mich freuen, würde die Mitarbeit auch in Zukunft so bleiben. Leider hat sich noch niemand für die englische Zusammenfassung gemeldet ... überlegt's Euch doch nochmal.

Artikel, die in dieser Ausgabe nicht enthalten sind, kommen in die nächste Ausgabe. Sollte ein Autor einen Artikel nicht finden, bitte ich um Reklamation: Ich bin nicht perfekt, bekomme Artikel auf Disk, per E-Mail, per Fax, in die Box und da kann unter Umständen mal etwas verloren gehen oder übersehen werden. Dies ist keine böse Absicht, Zensur oder dergleichen, also bitte nicht böse sein.

So, jetzt wünsche ich noch viel Spaß beim Weiterlesen dieser Ausgabe, bleibt mir nur noch der Wunsch, daß möglichst viele QLer mithelfen, daß das Österreich-Treffen im März ein Erfolg wird - Details auf der Rückseite dieser Ausgabe!

Euer

Jochen Merz



Grafikkarte & Monitor

Rolf Ritter

Mit diesem Artikel versuche ich ein wenig Klarheit über Grafikkarte und Monitor zu bringen.

1. Die Grafikkarte

Grundsätzlich hat die Grafikkarte die Aufgabe, mit oder auch ohne Hilfe des RAM den Bildschirminhalt (Monitorbild) zu speichern, um diesen dann dem Monitor in geeigneter Form weiter zu geben. Dieser Speicherbereich wird Bildschirmspeicher oder Videospeicher genannt. Da die Datenmengen, die im Bildschirmspeicher verwaltet werden, ein großes Volumen haben können und auch sehr schnell verändert werden müssen, wird dieser Speicher oft durch einen Grafikprozessor unterstützt. Außerdem sind natürlich sehr schnelle Speicherbausteine von Vorteil.

2. Bildschirmspeicher

Unser Computerbild wird durch ein Punktraster dargestellt. Als Beispiel die Aufteilung des MODE 4 Bildes des QLs. Es besteht aus 512 Punkten waagrecht (entspricht einer Zeile) und jeweils 256 Spalten senkrecht. Das sind 131072 Punkte (512 x 256). Jeder Punkt muß vier verschiedene Farben darstellen können. Diese Daten werden im Bildschirmspeicher abgelegt. Der Prozessor verändert also mit Hilfe des Bildschirmtreibers (Programmteil des Betriebssystems) bei Bildschirmveränderungen jedes entsprechende Speicherbit. Der Bildschirmtreiber ist somit die Software-Schnittstelle zwischen dem übrigen Betriebssystem bzw. Anwenderprogramm zur Bildschirmausgabe. Unabhängig davon muß die Grafikkarte den Bildinhalt in geeigneter Form 50 bis 100 mal pro Sekunde zum Monitor senden. Der Bildschirmspeicher hat in vielen Fällen ein eigenes RAM auf der Grafikkarte. Beim QL werden vom normalen RAM 32 Kilobytes abgezweigt. Aber wieso 32 Kilobytes? Wenn wir nur zwei Farben (Punktzustände) hätten, also Punkt leuchtet nicht (schwarz) oder leuchtet (weiß, grün oder auch bernstein), brauchten wir pro Punkt nur ein Bit (Bitzustand high oder low, entspricht an oder aus). Das wären dann $512 \times 256 = 131072$ Bits oder $131072 / 8 = 16384$ Bytes. In Kilobytes sind es dann $16384 / 1024 = 16$ Kilobytes. Das nennt man ein Monochrom-Bild (nur schwarz/weiß).

Zur Erinnerung: 1 Byte = 8 Bit, 1024 Bytes = 1 Kilobyte.

Aber wir brauchen noch mehr Speicher, da wir ja vier Farben darstellen wollen. Um vier Zustände (Farben oder beim Graustufenmonitor Graustufen) speichern zu können, werden zwei Bits benötigt ($2 \text{ hoch } 2 = 4$). Beispiel:

0-0 schwarz

1-0 grün

0-1 rot

1-1 weiß

Zwei Bit pro Punkt entsprechen also 131072 Punkte mal $2 = 262144$ Bits geteilt durch 8 also 32768 Bytes, geteilt durch 1024 gleich 32 Kilobytes Bildschirmspeicher.

Um 256 Farben bzw. Graustufen darzustellen, müssen wir pro Punkt selbstverständlich auch 256 Zustände speichern können. In 8 Bit ($2 \text{ hoch } 8 = 256$) können 256 Farben gespeichert sein. Wir brauchen also $512 \times 256 = 131072 \times 8 = 1.048.576$ Bits. Das sind ($1.048.576 / 8 = 131072$ Bytes / 1024 = 128 Kilobytes) 128 Kilobytes nur an Bildschirmspeicher.

Ein feineres Punktraster und mehr Farben bedeuten mehr Speicher, und somit bei gleicher Bildänderung mehr Speicheränderung. Bei gleicher Hardware-Qualität und höherer Auflösung wird die Grafikausgabe deshalb langsamer. Eine größere Bildqualität verlangt demzufolge mehr Rechnerleistung.

Hier noch ein paar Beispiele von Speicherbedarf und Bildschirmauflösungen:

1) für 512x256 Punkte:

Farben	Formel	Bits pro Punkt
4	2^2	2
8	2^3	3
16	2^4	4
256	2^8	8
65.536	2^{16}	16 (HiColor)
16.777.216	2^{24}	24 (True Color)

2) 800 mal 600 Punkte bei 65 536 Farben:

$800 \times 600 = 480.000$ Punkte

$480.000 \times 16 = 7.680.000$ Bits

$7.680.000 \text{ Bits} / 8 = 960.000$ Bytes

= 960 KBytes

3) 800 mal 600 Punkte bei 16,7 Millionen Farben

$800 \times 600 = 480.000$ Punkte

$480.000 \times 24 = 11.520.000$ Bits

$11.520.000 \text{ Bits} / 8 = 1.440.000$ Bytes

= 1,44 MBytes

4) Meine Auflösung mit der QVME und 15 Zoll Monitor: 768 mal 384 mit 4 Farben ergibt
 $768 \times 384 = 294.912$ Punkte
 $294.912 \times 2 = 589.824$ Bits
 $589.824 \text{ Bits} / 8 = 73.728$ Bytes

5) VGA (um DOS nicht zu vergessen): 640 mal 480 mit 16 Farben ergibt
 $640 \times 480 = 307.200$
 $307.200 \times 4 = 1.228.800$
 $1.228.800 / 8 = 153.600$ Bytes

Es werden nicht mehr als 16,7 Millionen Farben (True Color) benötigt, weil der Mensch nicht mehr Farben bzw. Graustufen unterscheiden kann. Bei vielen Grafikkarten können das Punktraster, die Farbtiefe und die Monitorparameter verändert werden. Bei der QVME - Karte kann man die möglichen Veränderungen durch BASIC-Befehle vornehmen. Die Bildschirmauflösung, die Bildfrequenz (Bilder pro Sek.) und die dazugehörigen Parameter müssen natürlich von dem Monitor darstellbar sein bzw. verarbeitet werden können. Der Monitor könnte sonst zerstört werden [**zu hohe Horizontal-Frequenzen sind gefährlich!**]. Im günstigstem Fall würde ein unscharfes Monitorbild entstehen. Also **VORSICHT** beim Programmieren ihrer Grafikkarte!

Wenn sie z.B. ihre QL-Auflösung (512 x 256) auf 1024 x 512 vervierfachen, werden natürlich nicht automatisch Ihre Programme angepaßt. Das heißt, daß z.B. XChange nach dem Starten sein Fenster mit 512 x 256 im oberen linken Eck aufbauen wird. Somit ist sein Fenster auf dem selben Monitor viermal kleiner. Viele Programme (QD) können ihre Fenstergröße verändern und somit die größere Auflösung nutzen. Der Vorteil der größeren Auflösung liegt aber auch darin, daß man mehrere Programme (Fenster) gleichzeitig durch Verschieben auf dem Monitor darstellen kann. Auch ein gepatchtes XChange läßt sich verschieben. Es sollte auch klar sein, wenn man eine Auflösung von 1024 x 512 benutzen will, muß man mindestens einen 17 Zoll Monitor zur Verfügung haben. Die Fenster werden sonst zu klein dargestellt und deren Inhalt (Schrift) ist kaum noch lesbar. Für die Auflösung 800 x 600 reicht nach meiner Meinung ein 15 Zoll Monitor.

Auf einem 15 Zoll Monitor lassen sich Bilder mit 1280 x 1024 Punkten noch betrachten, vorausgesetzt der Monitor kann die Auflösung verwalten. Dieses sollte man aber nur in Ausnahmefälle nutzen, da die Monitorfläche einfach zu klein ist.

Weiter gehts im nächsten Artikel mit den Monitoren.



Das QXL.WIN-Format

Teil 1

Gerhard Plavec

Das QXL.WIN-File Format ist ja recht praktisch, kann aber bisher leider nur unter SMSQ bzw. SMSQ/E und unter DOS - also nur mit der QXL-Karte oder QPC - verwendet werden. Das muß aber nicht so sein. Wer das Format kennt und versteht, kann durchaus auch von anderen Systemen auf die Daten zugreifen. Ich möchte mich daher mit dem Format befassen und Ansätze sowie Anregungen geben, wie man QXL.WIN-Files auch unter anderen Systemen verwenden könnte.

Zuvor erscheint es jedoch notwendig, einige Begriffe, wie "Clusters", "Map", "Sector", usw. möglichst verständlich zu erklären. Um das Problem richtig zu verstehen, gehen wir nicht von einem gegebenen Format aus, sondern überlegen uns, wie wir selbst eines schaffen würden. Nehmen wir also einmal an, wir wollen diverse Daten auf ein Medium speichern, für das es noch keinen Treiber gibt. Ein Treiber ist ein Programm, das sich nicht nur um die Kommunikation des Rechners mit dem Medium kümmert, sondern auch noch die Verwaltung und Organisation der Daten auf dem Medium übernimmt. Unser Medium ist vorerst vollkommen unformatiert, man kann es sich wie ein Eprom oder auch ein einfaches (leeres) File mit vorgegebener Größe und im Überschreibmodus vorstellen. Da es mir hier weniger um das Speichern, sondern vielmehr um die Organisation der Daten geht, nehmen wir einfach an, das Medium kann mittels eines einfachen Editors gelesen und (nur im Überschreibmodus) Be- und Überschrieben werden.

Als erstes wollen wir einige einfache Notizen speichern. Am einfachsten geht es, wenn wir sie schlicht hintereinander ablegen, vielleicht noch getrennt durch ein paar Sonderzeichen wie "*****" oder "=====", sofern diese in dieser Form nicht im zu speichernden Text vorkommen. Sucht man dann eine Notiz, genügt es, wenn man nach Trennzeichen sucht und die jeweiligen Notizen durchliest. Besser ist es allerdings, wenn man jeder Notiz einen Namen bzw. Titel gibt, das erleichtert dann die Suche erheblich bzw. ermöglicht auch eine Automatisierung des Suchvorganges. Die Schwächen dieser Methode erkennen wir jedoch, sobald eine Notiz gelöscht oder ergänzt werden soll: Da wir im Über-

schreibmodus sind, rücken die Daten in keiner Richtung nach. Dies würde ja bereits eine, wenn auch einfache, Datenverwaltung darstellen, die es für unser Medium noch nicht gibt. Das selbstverständliche Nachrutschen des Textes in einem Editor stellt für den Rechner ein recht aufwendiges Verfahren dar, es muß jeder einzelne Buchstabe des nachstehenden Textes umkopiert werden. Wenn es auch diverse Verfahren gibt, diesen Aufwand zu mindern, können diese nur im Hauptspeicher des Rechners durchgeführt werden, das Umkopieren auf dem Medium selbst wäre sehr zeitaufwendig. Man bedenke: Die Zugriffszeiten der schnellsten Festplatten nähern sich dem Millisekunden-, während sich jene der schnellsten RAM-Speicher dem Nanosekundenbereich, das ist immerhin ein Faktor von 1.000.000).

Zurück zu unserem nicht unterstützten Medium: Nach dem Löschen entstehen demnach "Löcher". Diese können bestenfalls mit gleichlangen oder unter Verlust mit kürzeren Notizen gefüllt werden. Um eine Notiz zu ergänzen bleibt uns nichts anderes übrig, als die Notiz ans Ende zu kopieren und den alten Platz als Loch freizugeben. Weiters müßten wir gegebenenfalls, um beim Lesen der Notizen keine Verwirrung aufkommen zu lassen, die "Löcher" mit Leerzeichen füllen, was wiederum mit Zeitaufwand verbunden wäre. Werden die Notizen oft geändert, entstehen zwangsweise immer mehr Löcher und die Suche nach geeigneten freien Plätzen gestaltet sich immer aufwendiger und wird zeitraubend, bis schlußendlich die Sache vollkommen unübersichtlich wird. In diesem Zusammenhang spricht man auch von einer Fragmentierung des Mediums.

Wir machen uns auf die Suche nach einer besseren Methode, dabei soll vor allem die Verwaltung der Löcher automatisiert und auch das aufwendige Umkopieren der Daten (beim Ergänzen) beseitigt werden. Wir unterteilen den Speicher in gleichgroßen Blöcken. Die Blockgröße wählen wir so "groß", daß die Mehrheit der Notizen darin reichlich Platz finden und daß der Speicher nicht allzusehr zersplittet wird. Wir müssen allerdings bedenken, daß es nicht mehr Notizen als Blöcke geben kann, was uns wiederum veranlaßt, die Blockgröße möglichst "klein" zu halten. Ein weiteres Argument, die Blöcke "klein" zu halten ist die Tatsache, daß der ungenützte Platz im jeweils letzten Block als verloren zu betrachten ist. Beispiel: Beträgt die Blockgröße 16KB, so können auf ein 20MB Medium maximal 1280 Notizen bzw. Files unter-

gebracht werden. Dieses Beispiel habe ich nicht zufällig ausgewählt, es entspricht den Gegebenheiten der ersten Harddisk, die es meines Wissens für den QL gab: CST-SCSI-Harddisk-Interface, das auch in so manchen Thor 8, 20 und 21 der gleichen Firma eingebaut war.

Die Größe der Blöcke richtet sich nach der Größe des Mediums und werden "Cluster" genannt, sie betragen im Allgemeinen ein Vielfaches eines "Sectors". Ein Sector besteht beim QL aus 512 Bytes.

Wie sind nun diese Clusters organisiert? Am Anfang zeigt ein Pointer (Zeiger) eines jeden Cluster auf den nächsten, am Anfang sind ja alle Cluster frei. Ein anderer Pointer zeigt, sobald etwas gespeichert wurde, auf den ersten besetzten Cluster und dessen Pointer wieder auf den nächsten besetzten Cluster. Es gibt also zwei Ketten: jene der freien und jene der besetzten Clusters. Jede Notiz belegt mindestens einen Cluster, wird sie gelöscht, wird der Cluster an die Kette der freien Cluster gehängt, wird sie ergänzt und benötigt hierfür einen weiteren Cluster, wird dieser aus der Kette der freien Cluster entnommen bzw. "um- und eingehängt".

Um nicht bei jedem Zugriff alle besetzten Cluster nach Trennzeichen und Namen durchforsten zu müssen, kann das System durch eine besondere Notiz, Directory (Inhaltsverzeichnis) genannt, ergänzt werden. Im Inhaltsverzeichnis werden all die Namen bzw. Titel der Notizen sowie ein Zeiger zum ersten Cluster der jeweiligen Notiz untergebracht. Dann können all die Trennzeichen entfallen und die Notizen durch Files aller Art ersetzt werden. Der Pointer der besetzten Clusters wird auch nicht mehr gebraucht, dafür brauchen wir aber einen, der auf den ersten Cluster des Hauptverzeichnisses zeigt.

Da die Länge des Files nicht automatisch ermittelt werden kann, muß diese irgendwo gespeichert werden. Weiters fallen noch andere Daten wie Filetyp, Dataspace und Datum an, die man gerne speichern möchte. Diese Daten betreffen zwar das jeweilige File, sind aber nicht Teil seines Inhaltes und werden in einem sogenannten "Header" (Kopf) gespeichert. Grundsätzlich ist es egal, ob man den Header beim File abspeichert, oder ob man die diversen Header in dem Verzeichnis (Directory) unterbringt.

Beim QL wird aber leider der Header - der jeweils aus 64 Bytes besteht - einerseits dem File vorangesetzt und andererseits aber auch dem Inhaltsverzeichnis einverleibt, so daß diese

JOCHEN MERZ SOFTWARE

Im stillen Winkel 12 • 47169 Duisburg • Germany
 ☎ 0203-502011 (Fax 0203-502012 Mailbox 0203-502013 & 502014)

Allgemeine QL-Programme

QD Editor	[V9.14]	DM 125,00
QMAKE	[V4.21]	DM 44,90
QLiberator SuperBASIC Compiler	[V3.36]	DM 139,00
QLoad-Ref	[V1.9]	DM 49,90
QLQ	[V1.13]	DM 69,90
QMAC Macro Assembler	[V1.01]	DM 69,00
QMENU	[V7.04]	DM 41,90
QPAC 1	[V1.05]	DM 61,50
QPAC 2	[V1.38]	DM 119,00
QTPY 2 Spell-Checker	[V2.17]	DM 82,50
QPTR Pointer Toolkit	[V0.30]	DM 89,90
QSpread Spreadsheet	[V1.42]	DM 169,00
QSUP	[V3.08]	DM 79,90
QMON/JMON	[V2.14]	DM 89,00
EPROM Manager	[V3.01]	DM 61,50
WINED	[V1.21]	DM 49,90
I/O 2 Toolkit	[V2.16]	DM 99,00
BASIC Linker	[V1.12]	DM 49,90
Fifi II Dateisucher	[V4.16]	DM 49,90
LDUMP	[V1.05]	DM 65,00
DISA Disassembler	[V3.02]	DM 95,00
DISA Upgrade from V1 or V2		DM 35,00
EasyPTR BASIC		DM 260,00
EasyPTR CAsm		DM 130,00
Stylus-Driver für text87 und text91		DM 69,00
HyperHelp for BASIC		DM 44,90
DiskMate 5		DM 69,00
CueShell		DM 95,00
SerMouse Treiber		DM 40,00
QDOS/SMS Reference Manual		DM 84,90
Update-Seiten von März 1997		DM 13,00
NEU: Update-Seiten von November 1997		DM 13,00

ProWesS & Anwendungen

Alle folgend aufgeführten Programme benötigen ProWesS, andernfalls laufen sie nicht!

ProWesS WindowManager-HTML Reader	DM 129,00
DataDesign Database	DM 79,00
NEU: LineDesign Vektor/DesktopPublishing	DM 79,00
PFlist	DM 49,00
fsearch	DM 49,00
NEU: fontutils	DM 79,00
NEU: PWfile	DM 64,00

QXL2 Karte mit 8MB RAM und der "erweiterten" Version von SMSQ namens **SMSQ/E** für nur **DM 799,-** oder ohne **SMSQ/E** nur **DM 659,-**
Es gibt wieder neue SuperGoldCards!!!
Bei Bedarf bitte kurz anrufen!

SMSQ/E für alle Systeme V2.85

SMSQ/E ist das neue Betriebssystem mit dem Sie Ihre QL-Programme laufen lassen können und eine Unmenge neuer Möglichkeiten erhalten: schneller, flexible Diskettenformate, viele und viel schnellere BASICs und sehr viel mehr!

Für QXL & QXL 2	DM 199,-
Für ATARIs mit QL-Emulator	DM 199,-
Für ATARIs ohne QL-Emulator	DM 249,-
Für GoldCard & SuperGoldCard	DM 199,-

QPC V1.41 mit SMSQ/E eingebaut!

Der QL-Emulator mit **SMSQ/E** läuft wunderbar! Man kann **SMSQ/E** auf PC's und Laptops ohne zusätzliche Emulator-Hardware laufen lassen! 486 oder Pentium ist Voraussetzung, ebenso MS-DOS 6 oder Windows 95.

Der Preis ist **DM 199,-** für Kunden, die **SMSQ/E** für andere Systeme bereits besitzen, andernfalls für ein komplettes System **DM 249,-**. Mit CueShell eingebaut nur **40,- DM** mehr!

... und zum Testen gibt's nun auch eine DEMO-Version, die alles macht bis auf Speichern - für nur DM 6,- incl. Disk & Porto!!!

Neuheiten!!!

- **QD** unterstützt in seiner neuesten Version auch wieder das **QDASM-Thing**.
- Sie finden **Jochen Merz Software** nun auch im Internet: <http://www.J-M-S.com/SMSQ/index.htm>
- Dazu gleich auch die email-Adresse: **SMSQ@J-M-S.com**
- Es gibt eine neue Version von **PageDesigner 3**: sie unterstützt auch wesentlich höhere Auflösungen! **SMSQ/E** ist Voraussetzung!
 Als neue Version: **139,- DM** - als Upgrade von vorherigen Versionen: **55,- DM**
- Auch neu: **Q-Route**. Reiseroutenplanung vom Feinsten! Mit vielen Fähigkeiten incl. grafischer Anzeige der Routen. Zur Zeit nur mit englischer Karte, an deutschen und europäischen Karten wird derzeit gearbeitet. **DM 89,-**
- Preissenkung bei der **Aurora**! Sie kostet nur noch **DM 399,-**. Grafikauflösungen bis **1024x512 Pixel**. Kann auch an modern **SVGA-Monitore** angeschlossen werden! Mit eingebautem Maus-Interface.

- **The Braquet**: vollständige Einbau-Plattform um **Aurora**s einfach in Tower-Gehäuse einzubauen. **DM 49,-**

- **QPC** gibt es nun in der Version **1.41**. Wichtigste Neuerung: es werden deutlich mehr aktuelle **VESA-Modi** der Grafikkarten unterstützt und damit auch Auflösungen von bis zu **1600x1200**. Zudem wird die **Bildschirmausgabe - je nach Grafikkarte - deutlich beschleunigt, sogar verdoppelt!** Auch wird der **Bildschirmmodus nach Nutzung der DOS-Box vollständig wiederhergestellt**.

Das Update ist nach wie vor **kostenlos** - **Originaldiskette mit Rückporto (3,- DM)** einschicken oder aus einer **JMS-Mailbox** ziehen! Die aktuelle Installation lassen Sie auf Ihrem Rechner, damit Sie weiter arbeiten können. Wenn Sie das Update erhalten, löschen Sie die Installation (**Achtung! Nicht die QXL.WIN-Festplatten-Datei löschen!**) und installieren neu.

Eine neue Anleitung zu **QPC** ist auch erhältlich. Hierin werden alle neuen Möglichkeiten von **QPC** erläutert. **Kostenpunkt: DM 10,-**.

QL Spiele

BlackKnight Schach	DM 119,90
Pipes	DM 29,90
BrainSmasher	DM 39,90
Arcanoid	DM 39,90
Firebirds	DM 39,90
QShang	DM 39,90
Diamonds	DM 39,90
The Oracle	DM 39,90
MineField	DM 39,90
Double Block	DM 39,90
The Lonely Joker 2	DM 59,00
SuperGamesPack	DM 90,00

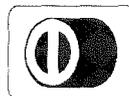
QL Ersatzteile

ZX8301	DM 19,90
ZX8302	DM 9,90
Tastaturfolie	DM 25,00

LIEFER- und ZAHLUNGSBEDINGUNGEN

Versandkosten [Deutschland] DM 8,80 (wenn Rechnungsbetrag unter DM 50,- dann nur DM 5,80). Bei Rechnungsbeträgen über DM 500,- kostet es

DM 18,- [Europa] DM 14,- (wenn Rechnungsbetrag unter DM 50,- dann nur DM 9,-). Alle Preise inkl. 15% MwSt. Irrtum und Preisänderung vorbehalten. Verrechnungs-, Euroschecks und Kreditkarten werden akzeptiert. Bankinzug möglich.



Daten theoretisch doppelt vorliegen müßten. Theoretisch - denn irgendwann hat man beschlossen, diesem doppelten Aufwand ein Ende zu setzen und es wird nur mehr jener Header unterstützt, der sich im Directory befindet. Den Daten jeden Files werden aber - vermutlich aus Kompatibilitätsgründen - nach wie vor 64 ungenutzte Bytes vorangesetzt. Jedenfalls darf man sich nicht wundern, wenn manche (ältere) Tools falsche Ergebnisse oder gar Fehlermeldungen liefern, z.B. das o.g. CST-Interface erwartet einen gültigen Header am Filebeginn, nicht nur bei der Harddisk, was sich nicht auswirkt, sondern auch bei Floppies.

Zurück zu unser Projekt. Die Pointer können jeweils im Cluster untergebracht sein oder auch gesammelt in einer Map (Übersichtskarte) untergebracht werden.

Daß die Unterteilung des Speichers in "Cluster" - oder wie die Fragmente auch heißen mögen - eine Vorgangsweise ist, die im Computerbereich weite Verbreitung findet und sogar die Grundlage für jede Organisation von Speicher und Daten darstellt, kann leicht nachgeprüft werden. Man findet ein solches System sowohl bei der Verwaltung des Speichers des QLs (Alchp-Bereich, Job-Bereich, Slaveblocks, etc) als auch bei den diversen Speichermedien wie Ramdisk, Floppy oder Harddisk.

Obwohl diese Systeme sehr ähnlich aufgebaut sind (Sectorengröße 512 Bytes, Header 64 Bytes, etc.) sind jedoch sehr genaue Kenntnisse notwendig, um sie verwalten zu können. Leider stehen einem aber nicht immer die notwendigen Unterlagen zur Verfügung.

So erging es mir auch beim QXLWIN-Format, trotz Hinterfragen konnte ich keine genauen Angaben erhalten. Jochen Merz schlug vor, doch einfach ein QXLWIN-File von der Formatierung bis zum Vollschieben zu beobachten, dann werde ich schon dahinterkommen, wie das Format aufgebaut ist. So geschah es dann auch. Hier die Früchte meiner Nachforschungen. Einige Punkte konnte ich nicht klären, sollte jemand mehr wissen, dann ersuche ich um Bekanntheit (E-Mail: gplavec@aol.com).

Das QXLWIN-File weist gleich am Anfang einen Haupt-Header (64 Bytes) auf, indem alle Daten (wie Name, Größe und freier Platz, aber auch Pointer (Zeiger) zu Cluster, etc über das Medium enthalten sind (*siehe Aufstellung in Tabelle 1). Anschließend kommt eine Zeiger-Tabelle aus der die Position der jeweiligen nächsten Cluster berechnet werden können und schließlich kommen die Daten, angeführt vom

ersten Cluster des Hauptverzeichnisses. Ein Verzeichnis (Directory) besteht aus jeweils 64 Bytes Einträgen (=File-Header), jedes Subdirectory (Unterverzeichnis) ist ein eigenes File (und als solches im übergeordneten Verzeichnis eingetragen, File-Typ ist 255) mit 64 Bytes Header gefolgt von 64 Bytes Einträgen usw. Im ersten Cluster eines jeden Files ist ein Header (64 Bytes) zu finden, der jedoch nicht verwendet wird. Die Daten des File-Headers befinden sich jeweils im 64 Byte-Eintrag des Directories (*s. Aufstellung in Tabelle 2).

Als Anregung schließe ich ein kleines Basic-Programm an, das jeder für eigene Zwecke weiterentwickeln kann. Es kann lediglich sämtliche Verzeichnisse auslesen und am Bildschirm anzeigen oder in ein File schreiben, dafür ist es aber übersichtlich und leicht nachzuvollziehen.

Es sollte jedem, der über minimale Programmierkenntnisse verfügt, leicht möglich sein, das Programm so umzuschreiben, da es auch einzelne Files ausliest bzw. kopiert. Insbesondere muß Zeile 500 adaptiert werden. Die gewählten Files wären dann je nach Typ und nach Erfragen des Zielnamens und Verzeichnisses, mittels SBYTES oder SEXEC abzuspeichern.

Leider läuft das Programm derzeit nur am QemuLator (QL-Emulator für den Mac) problemlos. Am QLAY (QL-Emulator für PCs, aber nicht zu verwechseln mit QPC) scheitert es daran, daß bei größeren QXLWIN-Files die Channel-Befehle GET, BGET aber auch INPUT oder INKEY\$, von Anfang an einen Fehler "End-of-File" bzw. "Datei-Ende" ausgeben. Ich versuche derzeit den Autor zu einer Korrektur zu animieren, vielleicht ist es bis zum Erscheinen dieses Beitrages schon soweit. Bei kleinen QXLWIN-Files geht es zwar, ist aber nicht besonders interessant.

Inwieweit das Programm mit UQLX (QL-Emulator für UNIX-Systeme) und auf anderen Systemen läuft, konnte ich nicht testen. Bedingung ist, daß ein QXLWIN-File unter S(uper)Basic mit OPEN geöffnet und mit den gewöhnlichen File-Bearbeitungsbefehlen bearbeitet werden kann.

Günstiger wäre es sowieso, einen Treiber zu schreiben, der es erlauben würde, die üblichen S(uper)Basic-Befehle zu verwenden. Nach meinen Kontakten mit Phil BORMAN (der die Treiber für das QUBIDE-Interface geschrieben hat) zu schließen, könnte bald ein neuer Treiber entstehen, der es erlauben würde QXLWIN-Files von Zip- und Syquest-Medien und vielleicht sogar

auch von CD-ROMs auf einem QL mit QUBIDE-Interface zu verwenden.

Eine weitere Anregung, die mir von Karl FLICKER zur Verfügung gestellt wurde, möchte ich auch niemanden vorenthalten. Es ist ein C-Programm, das unter DOS zu Compilieren ist und uns ermöglicht, unter diesem weitverbreiteten System einzelne QL-Files aus dem QXL-WIN-Konglomerat zu extrahieren und als DOS-File abzuspeichern. Auch dieses Pro-

gramm ist an sich nicht salonfähig, zeigt aber auf, wie ein QXL.WIN-File bzw. dessen Inhalt zu behandeln ist. Deren Ansätze können in eigenen Weiterentwicklungen verwendet werden.

Diese Beispiel-Programme und deren Sourcen (und Weiterentwicklungen) können auch im Internet unter <http://members.aol.com/KuEIWien/Programme.htm> bezogen werden.

Tabelle 1 / Main Header:

+00	long	"QLWA" (Erkennungsbegriff)
+04	word	length of name (Namenlänge)
+06	20 bytes	ASCII = Name
+1A	word	0000?
+1C	word	random number (fixe Zufallszahl)
+1E	word	access counter (Zugriffszähler)
+20	word	0000?
+22	word	number of blocks (512 bytes) in cluster (Clustergröße: 4 => 2048 Bytes)
+24		0000 0000 0000 ???
+2A	word	total sectors (Anzahl der vorhandenen Sektoren)
+2C	word	free sectors (Anzahl der freien Sektoren)
+2E	word	??? if total sectors = xx yy then yy xx+1(Kontrollzahl)
+30	word	0001 ?
+32	word	pointer to first free cluster (pointer x cluster = adress) (Zeiger zum ersten freien Cluster)
+34	word	pointer to main directory cluster (pointer x cluster = DIR) (Zeiger zum Hauptverzeichnis)
+36	long	length of main directory incl. header (Länge des Hauptverzeichnisses mit Header)
+3A		0000 0000 0000

Linked cluster map (Cluster-Tabelle):

+0040	word	pointer to next cluster or 0000 if end
+0042	word	pointer to next cluster or 0000 if end
...		
+xxxx	word	same until all clusters are pointed

Beispiel zum Auslesen des Hauptverzeichnisses (QXL.WIN-File mit etwa 20M, Clustergröße: h800):

+0034:	001A	----> x h800 = Adresse hD000 = DIR
nun schaut man unter (001A x 2 + h40 = 0074) wenn es 0000 wäre, hätte das Hauptverzeichnis keine weiteren Cluster, in unserem Beispiel aber schon, daher:		
+0074:	1939	----> x h800 = Adresse hC9C800
nun schaut man unter (1939 x 2 + h40 = 32B2) noch immer nicht 0000, weiter		
+32B2:	2605	----> x h800 = Adresse h01302800
nun schaut man unter (2605 x 2 + h40 = 4C4A)		
+4C4A:	0000	keine weiteren Cluster, Ende des Hauptverzeichnisses

So, im nächsten Teil geht es weiter mit den restlichen Strukturen und den Programmen.



Q40 - Projekt eines modernen QL

Peter Graf

Im Editorial zur deutschen Ausgabe 3 der QL Today schreibt Jochen Merz zur Lage der "kleinen" Computer. Er zeichnet ein düsteres Bild der zunehmenden Monopolbildung und ihrer Folgen, verbunden mit dem Aufruf mitzuhelfen, daß der QL als Alternative erhalten bleibt. Deshalb möchte auch ich mich seiner Bitte um Artikel nicht verschließen, und über eine neue Hardwareentwicklung berichten - den Q40.

Bis zum QL-Treffen 1997 in Solms ist über das Projekt "Q40" nichts nach außen gedrungen. Deshalb erst mal eine Beschreibung, worum es überhaupt geht.

Der Q40 ist ein komplett neues Mainboard, welches die QL-Hardware nachbildet oder ersetzt. Es enthält einen Motorola 68040-Prozessor mit einer kompletten Umgebung einschließlich High Color Grafik, Tastaturanschluß, Sound und Erweiterungs-Steckplätzen.

Die Platine paßt direkt in ein Industriestandard-Gehäuse und wird einfach an dessen Stromversorgung gesteckt. ROM und RAM haben einen 32 bit breiten Datenbus. Es können DRAM-Module (PS2-SIMMs) bis 32 MB eingesetzt werden, zwei Steckplätze sind vorhanden. Eine schnelle 32 bit Grafik direkt auf dem Mainboard bietet 65536 Farben bei einer Auflösung von bis zu 1024 x 512 Pixeln, stellt aber auch die QL-Modi im originalen Adreßbereich zur Verfügung.

Weiter gibt es einen Port für MF-II Tastaturen [**also gängige PC-Tastaturen - Editor**], Digital-/Analog-Wandler für Stereo-Sound, eine batteriegepufferte Echtzeituhr und ein 2 KB-RAM, das auch nach dem Ausschalten seinen Inhalt behält.

Das Mainboard trägt eine kompakte IDE/IO Karte als Interface zu 2 HD-Floppies, 2 IDE-Harddisks, 2 serielle Schnittstellen, Parallelport und Joystick. Es gibt zwei Erweiterungs-Steckplätze mit 16 Daten- und 20 Adreßleitungen. Einer davon ist mit der IDE/IO-Karte belegt.

Das Q40-Projekt ist nach jahrelanger Entwicklungsarbeit Wirklichkeit geworden. Was in Solms noch eine teilbestückte Platine war, ist nun ein Prototyp, der schon einen beachtlichen Teil seiner Hardwarefunktionen unter Beweis gestellt hat. Doch erst etwas zur Entstehungsgeschichte:

Als Sinclair-Fan der ersten Stunde habe ich mit ZX-81 und Spectrum angefangen. Einen QL besitze ich zwar erst seit kurzem, habe aber über meinen Bruder (Claus Graf) den gesamten Werdegang dieses Gerätes mitverfolgt. Vor etwa 7 Jahren habe ich mal eine 68020 Beschleuniger-Karte für den QL gebastelt, die auch funktionierte, aber aus Geldmangel nie zur Serienreife gelangte. [**Sehr schade!**]

Das QL-Betriebssystem war seiner Zeit voraus. Das, was dank Tony Tebby und anderen daraus wurde, ist modern und an Kompaktheit und Effizienz kaum zu überbieten. Bei der QL-Hardware sieht es etwas schwieriger aus. Man kann heute vielleicht zwei Hauptlinien unterscheiden. Erstens die Emulation in Hard- oder Software auf Fremdsystemen (z.B. QXL, QPC). Zweitens die QL-Erweiterungen, die entweder einen Original-QL oder Teile daraus benötigen (z.B. SuperGoldCard, Aurora). Mir persönlich fehlten in den letzten Jahren als dritte Linie die kompakten Komplettsysteme, die den QL ablösen könnten.

Irgendwann faßte ich den Entschluß, selbst ein neues Mainboard zu entwickeln. Daß es sich um ein Wahnsinns-Unterfangen handelt, so etwas nebenher in der Freizeit anzugehen, war mir bewußt. Aber es war ein langgehegter, persönlicher Wunsch. Hinzu kam, daß ich auch das langsame Aussterben anderer 680x0 Systeme befürchte.

Bei der Konzeption erschienen mir folgende Punkte besonders wichtig:

1. Schnelle Grafik mit vielen Farben für moderne Monitore
2. Schnelle CPU
3. Einfacher mechanischer Einbau und elektrischer Anschluß
4. Anschluß für IDE-Festplatten oder CD-ROM Laufwerk
5. Gängige Speichermodule, Erweiterbarkeit
6. Anschluß für eine moderne Tastatur

In den Detailfragen gab es viele schwierige Entscheidungen zu treffen und Kompromisse zu schließen. Nehmen wir zum Beispiel die Tonerzeugung. Der eine braucht gar keinen Sound, der andere nur ein einfaches Piepsen, der nächste will Hifi-Qualität. Ich habe mich dafür entschieden, die Ausgabe über zwei 8 Bit Digital-/Analog-Wandler mit wählbarer Abtastrate von 10 oder 20 kHz vorzunehmen. Damit erreicht man zwar keine Hifi-Qualität, aber eine einfache Handhabung, da kein speziell zu

programmierender Sound-Chip eingesetzt wird und es bieten sich doch mehr Möglichkeiten als beim Original-QL.

Ein weiteres Beispiel ist die Festlegung der Erweiterungs-Steckplätze. Einerseits hätte ich aus nostalgischen Gründen gerne den QL-Slot beibehalten. Andererseits hat der 8 Bit Datenbus des QL Nachteile beim Betrieb von schnellen 16 Bit IDE-Festplatten und es fallen mir kaum noch käufliche QL-Erweiterungskarten ein, die man sinnvoll im Q40 verwenden könnte. Es galt, einen 16 Bit Erweiterungsbus festzulegen. Der ISA-Bus mit seinen widersprüchlichen Timings und teilweise unsinnigen Signalen sollte es nicht sein. Also wurde als Kompromiß ein Q40-Erweiterungsbus definiert, mit wenigen Steuersignalen und wohldefinierten, unkritischen Timings, aber dem Steckverbinder des ISA-Bus. So lassen sich manche ISA-Karten in dem Q40 verwenden, aber ein Hardware-Bastler kann auch leicht selbst eine eigene Erweiterung bauen. Ein weiterer Vorteil dieses Konzepts ist, daß der Laufwerk-Controller-Chip nicht unmittelbar auf dem Mainboard sitzen muß - stattdessen wird kostensparend eine IDE/IO-Karte aus industrieller Massenproduktion eingesetzt, die einen ähnlichen Chip verwendet.

Das Q40 Mainboard besitzt eine Größe von ca. 16 x 20 cm und ist damit recht kompakt, wenn man bedenkt, daß auch High Color Grafik, Sound und die Erweiterungs-Steckplätze mit darauf sind. Der Q40 nimmt Abschied vom QL-Gehäuse, um dafür eine solide Einbaumöglichkeit in ein Industriestandard-Gehäuse zu bieten. Es sind lediglich vier Schrauben zu drehen, die Kabel und die IDE/IO-Karte einzustecken. Das entspricht sicher den Wünschen vieler QL-User, wenn auch der Abschied von der "good old black box" etwas wehmütig stimmt.

Am 8. November wurde auf dem QL-Treffen in Eindhoven der erste Q40-Prototyp in Aktion vorgestellt. Gezeigt wurden High-Color Grafiken in den Auflösungen 512 x 256 und 1024 x 512, die QL-Bildschirmmodi, Zugriffe auf eine IDE-Festplatte, Echtzeituhr, SER-Port und anderes. Da die Anpassung von SMSQ/E noch nicht begonnen hat, wurde hierzu ein spezielles ROM verwendet.

Die Resonanz beim Publikum war unerwartet hoch. Unser Stand in Eindhoven war so umlagert, daß wir den ganzen Tag nicht eine ruhige Minute bekamen und kaum mal einen Schluck trinken oder essen konnten.

Nachmittags hatten wir Gelegenheit, mit Tony Tebby einige technische Details zu besprechen.

Die zur hardwarenahen Programmierung benötigten Eigenschaften des Q40 Mainboards wie Interruptbehandlung, Register und Speicherbelegung werden übrigens für jedermann offengelegt und sollen nicht einigen wenigen vorbehalten bleiben.

Zum Schluß noch etwas über den Stand der Entwicklung: Ein Redesign der Platine in Multi-layer-Ausführung findet gerade statt, danach kann in Kürze die Anpassung von SMSQ/E starten. Außerdem wird eine Variante mit 68060 CPU vorbereitet.

Hierzu noch eine interessante Nachricht von Motorola: Im April 1998 soll die Produktion der 68060 CPU in einer neuen 0,42 Micron Fertigungstechnik starten. Dadurch sollen schnellere Timings mit Taktraten von mindestens 75 MHz ermöglicht werden. Nachdem Motorola lange Zeit die 680x0 Linie vernachlässigt und durch unangemessen hohe Preise benachteiligt hat, ist dies wirklich ein Lichtblick.

Kontaktadresse:

Dipl.-Ing. Peter Graf

Lahnstr. 32

35239 Steffenberg

Email: pgrafq@t-online.de

Anmerkung des Editors: Die Q40 hat tatsächlich große Begeisterung ausgelöst. Von allen übrigen Vorteilen abgesehen gibt es einen unschätzbaren Vorteil: es gibt sie und sie läuft! Wer sie in Eindhoven nicht bestaunen konnte sollte gleich noch einmal einen Blick auf's Titelbild dieser Ausgabe werfen.

Nun fehlt es nur noch an der Software, dann haben wir endlich den lang ersehnten, "richtigen" QL-Nachfolger.



Software-Updates über's WWW

Jochen Merz

Einige Kunden hatten ja schon nachgefragt, wie es sei, Software-Updates über's Internet bereit zu stellen. Nun, das wird in absehbarer Zeit sicherlich nicht per FTP-Download passieren, weil für mich einfach viel zu viele Faktoren dagegen sprechen. Es gäbe aber eine andere Alternative....

Warum ich dagegen bin werde ich zuerst darlegen, dann wird auch verständlich, warum mir die andere Möglichkeit wesentlich besser gefällt.

Die kostenlosen Updates über die Mailboxen bleiben bestehen. Nicht jeder hat Internet-Zugang, und daher kann ich durch die Mailboxen mehr Kunden den Service bereitstellen.

Ein Update-Service über's Internet wäre also zusätzliche Arbeit, für die ich weitere Zeit (die bei mir ohnehin sehr knapp bemessen ist) finden müßte. Bei Internet-Downloads ist es so, daß ich neue Versionen auf meine Kosten auf den Server packen müßte ohne zu wissen, ob sie überhaupt geladen werden - und dies wäre doch pure Verschwendung.

Zudem unterliegt der Internet-Server keiner Kontrolle, wie ich sie über die Mailbox hier habe, d.h. ich kann keinem Kunden gezielt Zugang zu bestimmten Bereichen ermöglichen. Passwörter sind von mir nicht kontrollierbar, und es wäre zusätzlicher Aufwand. Zudem: Server-Platz kostet Geld!

Nun bekam ich den Vorschlag, die Updates doch an "abonnierte" Kunden per E-Mail zu versenden. Jeder bekäme dann automatisch die neueste Software-Version zugestellt, ohne ständig nachschauen zu müssen. Ich wüßte, das meine Updates auch genutzt würden, also keine verschwendete Zeit darstellen, und zudem scheint der Versand von E-Mails wesentlich effizienter zu sein als Seiten und Dateien über's WWW (WeltWeites Warten) anzufordern.

Die Idee gefällt mir, und wenn genügend Interesse besteht, würde ich diesen Service wahrscheinlich auch einrichten. Ich habe dazu einen Fragebogen (in Englisch) auf meiner Webseite bereitgestellt, den Interessenten bitte ausfüllen.

Je nach Ergebnis kann der E-Mail-Update-Service bald Realität werden.



Uhren-Korrektur

Dietrich Buder

1. Allgemeines

Im Zeitalter von Funkuhren ist eine Abweichung von mehr als einer Minute fast schon wie ein Sakrileg. Bei mancher GoldCard oder Super-GoldCard jedoch tritt eine derartige Abweichung schon nach drei oder gar zwei Wochen ein. Das ist zweifellos nicht der Stand der heutigen Quarzuhren-Technik. Schon in 1996

sprach ich deshalb Stuart Honeyball von Miracle diesbezüglich an und fragte, ob er nicht einen Abgleich mit einem Trimmer veröffentlichen könne, damit ich meine Uhren entsprechend umbauen kann. Er nahm meine Anregung zur Kenntnis und teilte mir letztes Jahr auf dem QL-Treffen in Solms mit, er mag einen Hardware-Umbau nicht und bevorzuge eine Software-Lösung. Mir gefiel diese Aussage zwar nicht so ganz und mir war auch nicht klar, ob es keine "saubere" Trimmer-Lösung auf der Oberseite der Karte gibt oder ob er den Lötkünsten der QLer misstraut. Gerade für Brikett-QLs wäre die Hardware-Lösung vorteilhaft, weil ja nicht immer mit der gleichen Disk gebootet wird.

In QL Today Band 1 Heft 6 folgte schließlich eine kurze Beschreibung seines Programms und die Ankündigung einer Veröffentlichung auf der ersten Coverdisk. Durch irgend ein Versehen fehlte aber diese Datei auf der Disk und deshalb kam in QL Today Band 2 Heft 1 eine Entschuldigung und der fehlende Ausdruck seines Programms.

Im Band 2 Heft 3 folgte dann noch ein Uhren-Korrektur-Programm von Ian Pizer, das sehr schön kurz und einfach ist und damit für mich leichter zu studieren und zu tippen war.

Das Ergebnis meiner Untersuchungen ist: Das Programm von Ian Pizer funktioniert, ist aber theoretisch etwas ungenauer und ohne jeglichen Komfort. Das Programm von Stuart Honeyball ist extrem komfortabel und somit außerordentlich aufwändig, kann aber nur unter SMSQ/E laufen. (Ich hatte keine Lust, es abzutippen und ggf. zu übersetzen, auch mag ich seine ausgeschriebenen Variablen nicht.)

Damit erhebt sich für mich eine grundsätzliche Überlegung. Wie soll man es mit QDOS und SMSQ/E halten. Ich persönlich kann aus früher schon genannten Gründen nicht auf QDOS verzichten und schreibe deshalb meine QL-Programme generell für dies Betriebssystem; sie funktionieren dann auch unter SMSQ/E. Für Programme, die nur auf der QXL laufen können, gilt das natürlich nicht. So kann ich nun eine Lösung vorstellen, die hinsichtlich Genauigkeit der von Stuart entspricht, im Komfort und Länge einen Mittelweg darstellt und vor allem auch unter QDOS läuft.

2. Arbeitsweise

Das Korrektur-Programm sollte an einer sinnvollen Stelle in das BOOT-Programm eingebaut werden. Wenn das erfolgt ist:

- Die Abweichung der Uhr innerhalb einer Woche ermitteln oder schätzen.
- Das betreffende BOOT-Programm laden.
- Prozedur 'UHR' im Direktmodus aufrufen (Zeile 2980).
- Die Sekunden zur Korrektur nach genau einer Woche eingeben (Zeile 3000), ggf. sogar Dezimalbrüche

Das Programm legt dann die Datei UHR_DAT an und speichert darin das genaue Datum der Eingabe (siehe Abschnitt 3) sowie die notwendigen Korrektur-Sekunden pro Woche (Zeile 3020). Hierbei habe ich das Schema von Ian Pizer übernommen. Genauso gut ginge auch die Abweichung der Uhr pro Woche; dazu müssten nur Text und Vorzeichen geändert werden.

Bei jedem Booten werden in der Prozedur UHRKORR die gespeicherten Daten der Datei UHR_DAT eingespielt (Zeile 2820) und aus der verflossenen Zeit sowie der eingegebenen Sekunden pro Woche eine aktuell notwendige Korrektur k berechnet (Zeile 2840). Wenn dies Ergebnis mehr als vier Sekunden beträgt (Zeile 2850), wird die Uhr nachgestellt (Zeile 2920) und das Datum der Korrektur in die Datei UHR_DAT übergeben (Zeile 2940).

Der Komfort meines Programmes besteht darin, dass ich vor einer Korrektur die verflossenen Stunden (Zeile 2870) sowie die berechnete Korrektur k anzeige (Zeile 2890) und danach die noch ungenau gehende Uhr starte (Zeile 2910). Anhand einer Funkuhr lässt sich somit leicht überprüfen, ob die bisherige Abweichung der Uhr noch stimmt bzw. korrigiert werden muss. Anschließend wird die nachgestellte Uhr zur Kontrolle nochmals gestartet (Zeile 2260).

Ian Pizer hat auf eine Prozedur UHR verzichtet und arbeitet im Direktmodus (Zeile 3020). Bei der Umstellung auf Sommerzeit bzw. Winterzeit, wenn die Abweichung nicht mehr stimmt und auch bei der Ersteingabe erleichtert aber die Prozedur UHR die Eingabe.

3. Aktuelles Datum

Das aktuelle Datum kann mit PRINT DATE\$ oder PRINT DATE angezeigt werden. Beim Schreiben dieser Zeilen erhielt ich: 1997 NOV 06 22:00:24 und 1.16285E9. Die erste Angabe ist klar, die zweite Zahl gibt die Sekunden an, die seit dem 1.1.1961 0 Uhr verflossen sind. Im Jahr 1997 ist dies eine 10-stellige Zahl. Ausgedruckt werden aber maximal nur sieben Stellen. Die nicht angezeigten Stellen lassen sich beim QL wie bei jedem Taschenrechner durch Subtraktion einer passenden Zahl der ersten Stellen

ermitteln. In meinem Beispiel mit 1.16285E9:

`d=DATE: PRINT d, d-1.162E9` ergibt

1.16285E9 und 850424, also

`d=1.162850424E9=1162850424` Sekunden seit dem 1.1.1961.

Wenn der gleiche Befehl nach wenigen Sekunden wiederholt wird, erhöhen sich nur die letzten Stellen der zweiten Zahl um eben diese Sekunden. - Das ist auch der Grund, warum das Programm von Ian Pizer theoretisch etwas ungenauer arbeitet, da er nur das 7-stellige DATE verwendet. Sein aktuelles Datum hat im ungünstigsten Fall einen Fehler von 999 Sekunden oder rund 17 Minuten, was in der Praxis vermutlich zu vernachlässigen ist.

Hier zeigt das Programm von Stuart Honeyball eine elegante Lösung, die ich im Prinzip übernommen habe. Er spaltet das 10-stellige DATE auf mit:

`d=DATE: d1=d DIV 65536: d2=d MOD 65536`

Leider laufen DIV und MOD bei so großen Zahlen nur unter SMSQ/E. Bei QDOS erscheint die Fehlermeldung ÜBERLAUF. Erfreulicherweise funktioniert aber unter QDOS der Taschenrechner, am einfachsten mit einer Hilfszahl h :

`d=DATE: h=d/65536: h1=INT(h): h2=h-h1` (Zeilen 2930 und 3010)

Die Zahlen $h1$ und $d1$ sind gleich und stellen die Vorkomma-Stellen der Hilfszahl h dar. Die Zahl $d2$ ist der Rest aus der Division mit DIV, $h2$ jedoch sind die Nachkomma-Stellen der Hilfszahl h . In meinem Programm werden $h1$ und $h2$ in der Datei UHR_DAT abgelegt, bei Stuart entsprechend $d1$ und $d2$.

Das Zusammensetzen zu dem ursprünglichen 10-stelligen DATE d erfolgt umgekehrt mit:

`d=(h1+h2)*65536` (Zeile 2830) bzw. bei Stuart `d=d1*65536+d2`

4. Zusätzliche Bemerkungen

Ein grundsätzliches Problem sei nicht verschwiegen: Einfache Quarzuhren sind leider temperaturabhängig und spannungsabhängig. Ich kenne nicht die genaue Stromversorgung der Uhr, habe aber den Verdacht, dass sie bei eingeschaltetem QL vom Netzteil gespeist wird. Speziell bei schwacher Batterie sind deshalb die Abweichungen abhängig von der jeweiligen Benutzungsdauer des QLs.

Weiterhin fand ich eine Ungereimtheit, die ich nicht erklären kann und als gegeben hinnehme: In Zeile 3020 muss als Dateiname 'win1_UHR_DAT' stehen. Mit `lq$&'UHR_DAT'`, was ich bevorzugt hätte, gibt es manchmal eine Fehlermeldung.

Ian Pizer hat in sein Programm PROT_DAT 0 und PROT_DAT 1 vor bzw. hinter die Uhrverstellung ADATE+k eingebaut. Laut GoldCard-Handbuch stehen GoldCard bzw. SuperGoldCard beim Einschalten auf PROT_DATE 0. Es ist also nur PROT_DATE 1 nach der Uhrkorrektur erforderlich (Zeile 2270), wenn nach dem Booten ein Verstellen der Uhr verhindert werden

soll. Falls das Programm grundsätzlich auch auf der TrumpCard oder SandyCard laufen soll, gibt es eine Abfrage für die GoldCards

```
ka$=CHR$(PEEK(65546)) und dann
```

```
IF ka$='G':PROT_DATE 1
```

Ich meine, eine weitergehende Beschreibung des Programms ist nicht nötig. Ich wünsche allen Nutzern viel Erfolg damit.

```
1040 :
1050 lq$='win1_'
1060 :
2250 :
2260 UHRKORR: EW lq$&'QL_UHR_EXE': REMark Uhrstart nur mit EW !
2270 PROT_DATE 1
2280 :
2780 :
2790 DEFine PROCedure UHRKORR
2800 REMark Uhr-Korrektur nach Jan Pizer aus QL Today Heft 2/1997 Seite 38
2810 REMark Prozedur UHR nach korrekt gestellter Uhr einmal im Direktmodus starten
2820 OPEN #3;lq$&'UHR_DAT': INPUT #3;h1,h2,sec: CLOSE #3
2830 d=(h1+h2)*65536
2840 z=DATE-d: k=INT(sec*z/604800)
2850 IF ABS(k)>4
2860 BEEP 800,60
2870 AT 6,15: PRINT 'Vor'!INT(z/3600)!'Stunden war die letzte Uhren-Korrektur.'
2880 IF sec>0: a$='vor': ELSE a$='zurück'
2890 AT 9,15: PRINT 'Die Uhr wird'!ABS(k)!'Sekunden'!a$;'gestellt.'
2900 INK 4: AT 20,40: PRINT 'Beliebige Taste drücken !': INK 7
2910 PAUSE: CLS: EW lq$&'QL_UHR_EXE'
2920 ADATE +k
2930 d=DATE: h=d/65536: h1=INT(h): h2=h-h1
2940 OPEN #3;lq$&'UHR_DAT': PRINT #3;h1\h2\sec: CLOSE #3
2950 END IF
2960 END DEFine UHRKORR
2970 :
2980 DEFine PROCedure UHR
2990 CLS #0: PRINT #0;\,'Die Uhr muss vor [+] oder zurück [-] gestellt werden.'
3000 INPUT #0;,'Bitte notwendiges Stellen in Sekunden pro Woche eingeben: ';sec
3010 d=DATE: h=d/65536: h1=INT(h): h2=h-h1
3020 OPEN_OVER #3;'win1_UHR_DAT': PRINT #3;h1\h2\sec: CLOSE #3: CLS #0
3030 END DEFine UHR
3040 :
```

Versuch eines Berichts vom 6. Internationalen Spectrum-User-Club Treffen in Sindelfingen am 15.11.1997

Otmar Peters

(Ein etwas langer Titel für ein kurzen Bericht?)

Eine kurze Vorgeschichte

Am 14.11.97 kam (endlich) die neueste QL-Today Ausgabe. Auf der Rückseite der deutschen Ausgabe prangte die Ankündigung für das Sinclair Use. C... - Treffen. Eine Nachfrage bei Jochen Merz brachte mir keine weiteren Informationen, dafür den Auftrag, einen (diesen) Bericht vom Treffen zu schreiben.

Die Erwartungen

Wie hieß ihr Homecomputer vor dem QL? Ich hatte einen Spectrum mit 48k und fürchterlichen Gummitasten, trotzdem - die Erinnerungen an den Spectrum sind eher positiv.

Auf der Hobbyelektronik in Stuttgart eine Woche zuvor waren die C64-User mit einer sehr interessanten Entwicklungen vertreten (C64 mit SCSI-Interface 20 MHz Takt und 16 MB Speicher), so könnte mit dem Spectrum doch einiges geschehen sein.

Vielleicht treffe ich auch andere QLer ...

Das Treffen

Die Anreise mit S-Bahn und Bus bis Sindelfingen ging recht flott, die letzten Meter bis zum Hotel waren auch schnell geschafft. Gegen 11:00 Uhr kam ich im Hotel Residence an.

Eine freundliche Dame an der Reception schickte mich in das zweite Untergeschoss. Dort fand ich ca 10 SUC-ler in einem Raum, für die Zeit doch recht erstaunlich (???).

In der Mitte des Raumes war auf vier zusammengestellten Tischen das aktuelle Angebot an Soft- und Hardware aufgebaut. An den Wandseiten traktierten die Teilnehmer ihre Geräte; es klang wie in einer Spielhöhle, "And the Beep goes on".

Thomas Eberle, der SUC- Guru(!?), begrüßte mich herzlich und überreichte mir auch eine Ausgabe der Clubzeitschrift. **[Nein, pure Namensgleichheit - es handelt sich NICHT um den Herausgeber der Ex-68000er - Editor]**

Meine neugierigen Fragen bezüglich der SUC-Lage, der Hard- und Software-Entwicklung beantwortete er geduldig.

Die Mitgliederzahl des SUC liegt bei ca 200. Geräte sind noch genug vorhanden. Nachbauten des Spectrums aus Tschechien und Russland seien auch verfügbar. (Ein Gerät aus tschechischer Herkunft war auch für 80,- DM angeboten.)

Im ehemaligen Ostblock seien die Spectrum-User auch noch etwas stärker vertreten.

Spiele, Spiele, dies ist der Kitt, der die SUC-Gemeinde zusammenhält. Eine CD mit 3000 Spielen für den Spectrum gibt es auch. Auf der CD ist auch ein Spectrum-Emulator für den PC vorhanden (doch dazu später mehr). Der Hardware-Leckerbissen war ein Tastaturadapter, der das Gummitastenproblem des Spectrums (1) löst. (s.o)

Bemerkenswert fand ich die vielen älteren Zeitschriften, die zum Verkauf angeboten waren: Happy-Computer, Computer Kontakt, Sinclair User, alte Ct's; die Zeitreise in die 80er war sehr interessant.

Die Internationalität war keine Floskel, aus Holland und Österreich (Sinclair Kolonie?) waren Teilnehmer gekommen.

Die Teilnehmerzahl aus dem "Ländle" (Baden Württemberg) war, wenn man die Besucher aus dem Kölner Raum und aus den neuen Bundesländern berücksichtigt, nicht sehr erhebend. Während meiner Anwesenheit bis 14:00 waren maximal 17 Personen anwesend.

Die QLer sind wohl die Super-Schwaben und erledigen ihre Kehrwoche, bevor sie auf solch ein Treffen gehen (es hat sich von den Anwesenden sonst keiner als QLer oder SMSQ-Anhänger geoutet).

Nicht sehr schwer bepackt, außer mit einer Speccy CD, ging der Besuch in einer (doch) anderen Welt zu Ende.

Mein Fazit

Die "Spectrum Welt" ist sehr spielzentriert, die Hardware spielt eine sehr untergeordnete Rolle.

PS. Der Spectrum-Emulator von der CD ist aber wirklich stark. Ohne Software-Bremse sind die Spiele unbrauchbar. Der PAC-MAN ist wirklich lustig, wenn ich auch sonst kein Spiele-Freak bin.

Programmieren in Basic wäre mit dem Emulator von der Geschwindigkeit her möglich, doch ... (man ist halt zu verwöhnt).

Der Besuch war keine Zeitverschwendung und das ist doch auch schon etwas.

Weitere Infos bei

Thomas Eberle, Tel./Fax 0711-775033

Anmerkung des Editors: Prima, zeigt dieser Artikel doch, daß sich die QL-Gemeinde nicht verstecken braucht: besser besuchte Treffen, individuelle Anwendungen (nicht nur Spiele) und eine wesentlich größere User-Basis können uns alle nur positiv stimmen.

Schade, daß nicht mehr QLer zum Treffen gekommen sind - es war zugegebenermaßen ja sehr, sehr kurzfristig angekündigt und so manch Leser hat QL Today vielleicht auch erst nach dem 15. erhalten.

Wie dem auch sei - ein echtes QL-Treffen im Süden steht an - die Heft-Rückseite sagt alles!



Zusammenfassung der englischen Ausgabe 5

Jochen Merz

Neuigkeiten

TF Services meldet, daß die Hardware der RomDisq fertig ist und nur noch auf den Treiber gewartet wird.

QBranch bietet ein neues Backup-Programm namens "Knight Safe".

Jonathan Hudson hat Qascade erweitert und bietet auch ein PD-Programm namens "bruz" - kurz für "Backup and Restore Utility" - also Datensicherung und -Rettung. Außerdem hat er in Zusammenarbeit mit Christopher Cave ein vollständiges C-Interface für QMENU programmiert, das auch PD ist (und auch in den JMS-Mailboxen zu finden ist).

Albin Hessler hat seine Produktpalette gestrafft: EASYPTR Basic (vormals Teil 1 und 2) kostet nun 276,- DM, EASYPTR CAsm (vormals Teil 3) kostet nun DM 138,-. CueShell kostet ebenfalls DM 138,-. Im Bundle mit QPC bleibt es bei DM 40,- Aufpreis für CueShell. Anleitungen gibt es nur noch in Englisch.

Mark Knight berichtet über die von ihm geschriebenen PD-Programme, die alle in der Quanta-PD-Bibliothek zu bekommen sind.

Q40 - diesen Artikel finden Sie hier in dieser Ausgabe auf Deutsch.

Nasta, Designer von Qubide, der Aurora usw. schreibt in einem sehr ausführlichen Artikel über mögliche Lösungen bei Problemen mit Qubide. Seine Tips in Kürze:

- vernünftige Stecker, kurze Kabel.
- alle Spannungen ordentlich anschließen.
- nur ein Netzteil für alles verwenden.
- gutes Netzteil mit stabilen Spannungen nehmen.
- alle Bauteile direkt mit +5V versorgen (also die Spannungsregler 7805 umgehen).
- Störstrahlung von Laufwerken berücksichtigen.
- auch die von Zusatzkarten wie SuperGold-Card generierten Interferenzen berücksichtigen!
- Master-Slave Probleme (Friedemann Oertel hat dazu auch noch ein paar Tips in der englischen Ausgabe).

Ein sehr interessanter Artikel, den sich jeder, der ab und zu defekte Harddisk-Maps oder

ähnliches vorfindet, zu Gemüte führen sollte.

Jim Hunkins, der in der letzten Ausgabe von QL Today auf ProWesS einging, beschäftigt sich in dieser Ausgabe mit ProWesS-Anwendungen.

Da wären die Font-Utilities, die es Ihnen ermöglichen, Adobe Typ 1 Schriften zu konvertieren, so daß sie mit ProWesS genutzt werden können. FontShow zeigt Ihnen den gesamten Zeichensatz mit allen Details an, und FontPreview gibt eine Vorschau wie der Zeichensatz auf dem Bildschirm aussehen wird.

PFList gibt alle möglichen Arten von Text in sehr hoher Qualität aus. Hoch- und Querdruck ist möglich, ein- und zweispaltiger Druck ebenfalls.

LineDesign ist eine wohl bekannte PROGS-Anwendung, wobei die neueste Version ProWesS zwingend voraussetzt. Endlich können Sie auch echte Farbausdrucke bekommen! Das Update ist kostenlos, aber wie gesagt, ProWesS ist Voraussetzung. Da mehr und mehr ProWesS-Anwendungen erscheinen (eine Textverarbeitung ist auch in Arbeit, bzw. sogar zwei verschiedene!) lohnt sich die Anschaffung.

Es folgt der zweite Teil der Begriffserklärungen der gängigsten "Computer"-Abkürzungen.

Mit Byts of Wood started Roy Wood eine ab jetzt in jeder Ausgabe erscheinende Kolumne, in der er über Dies und Das in und aus der QL-Szene berichtet. Größere Themen sind:

- die QL-User E-Mail Newsgroup (schreibt an ql.users@nvg.ntnu.no, Betreff "subscribe").
- die seriellen Ports der Aurora (Port 1 und Port 2 sind vertauscht beschrieben).
- das November-Treffen in Portishead.
- das im Februar stattfindende QL-Treffen in Hove (bei Brighton).

Ein gesonderter Teil dieser Ausgabe widmet sich dem PD Pinboard. Hier beschreiben verschiedene Tester eines oder mehrere Public-Domain-Programme. Die Programme sind u.a. auch aus den JMS-Boxen zu beziehen.

Darren D. Branagh testet Frank Dibowskis tolle PD-Spiele Crazy Cards, Minefield, Einstein, 23 Bullets und Puzzel. Alle schneiden (verdienterweise) gut ab.

Dilwyn Jones beschäftigt sich mit dem CIA World Factbook.

Timothy Swenson beschreibt das tolle Utility **Qasqade** von Jonathan Hudson - sollte wirklich jeder haben: Kaskadierte Menüs mit denen man Programme starten kann (ähnlich wie bei Windows95) sind einfach zu erstellen.

Dilwyn Jones beschreibt einen sehr schönen **CD Player für QPC** - sieht aus wie ein Auto-CD-Player. Wem das QPC beigelegte CD-Player-BASIC-Programm nicht gefällt, der findet mit diesem CD Player eine sehr schöne Alternative.

Darren Branagh testet eine Menge der **PD Pointer Programme** von Phil Borman, die in der Quanta Bibliothek zu finden sind: Zeichensatz-Anzeige, Hex-Rechner, Datei-Viewer und verschiedene andere Programme wurden getestet.

Simon N. Goodwin beschreibt in seinem **"About Time"** eine weitere Lösung zur genauen Zeit-Bestimmung und geht auch damit das Problem an, das Dietrich Buder in seinem Artikel beschäftigte: Stuarts Programm läuft nur auf SMSQ. Simon korrigiert es, so daß es überall läuft und hat auch sonst noch ein paar interessante Anmerkungen zu diesem Thema.

In seinem dritten und letzten Teil über ergonomische Computer-Arbeitsplätze beschäftigt sich **Geoff Wicks** mit Computer-Bildschirmzeichensätzen, der Verwechselbarkeit von verschiedenen Zeichen wie O und 0, B und 8, Z und 2 und dergleichen. Er beschreibt auch Probleme bei der Auswahl von Farben und Kontrast.

John J. Impellizzeri erläutert den **Anschluß von 5.25" Diskettenlaufwerken an den QL** - wer immer noch Daten in diesem Format gespeichert hat findet hier u.U. Hilfe, wenn sein Laufwerk mal den Geist aufgibt.

In der **Briefkasten-Ecke** gibt es hilfreiche Tips zum Umgang mit ZIP-Laufwerken an QPC, seriellen Modem-Kabeln am QL und dem Thema Druckertreiber für HP Deskjet und Canon-Druckern [*zu problemlosen EPSON habe ich ja schon in der letzten Ausgabe etwas geschrieben*].

Wer bislang dachte, man kann DD-Disketten mit Bohrung auf HD "ummünzen" und mit kleinem Etikett aus HD eine DD-Disk machen, der liegt falsch! **Mark Knight** erläutert ausführlich, warum verschiedene Aufzeichnungs-Dichten verschie-

dene Köpfe in den Laufwerken benötigen und warum man Disketten nicht einfach tauschen sollte: Die Oberfläche ist mit speziellem Material beschichtet, das auf die Aufzeichnungs-Frequenzen abgestimmt ist. Wer seine Daten also nach ein paar Jahren noch lesen können möchte, sollte die richtigen Disketten nehmen.

Der dritte Teil der **logischen Grundlagen** von **Stuart Honeyball** beschäftigt sich mit einem BCD-Zähler, der von 0 bis 9 zählen kann.

Es folgt der zweite Teil des Artikels von **Herb Schaaf** über **QL-Grafiken** (man erinnere sich an die letzten englischen Titelblatt-Grafiken). Der fehlende Teil von Herbs Listing ist hier zu finden und auch eine Erläuterung zu seinem "Escher in Motion" Programm.

Friedemann Oertels Hardware-Abenteuer-Bericht geht ausführlich auf QXL-Kühlprobleme ein. Auch hier sind wertvolle Tips zum Anschluß von Festplatten-Kombinationen an Qubide zu finden, zudem noch generelle Tips zum Umgang mit Qubide. Abgeschlossen wird sein Artikel mit Hinweisen zum Betrieb des Falkenberg-T90-Interfaces.

Der letzte Artikel dieser Ausgabe ist eine Art **Show-Report** mit (im Testausdruck sehr gut aussehenden) Bildern - hoffen wir, daß sie so gut bleiben! **Simon N. Goodwin** berichtet über die Planung eines verbesserten ZX8301-Ersatzes, den Miracle und TF Services gemeinsam entwickeln wollen. Bildschirmausgaben sollen damit dreimal schneller sein als bisher (dies ist beim QL ja auch sehr langsam), zudem soll er bis zu dreimal höhere Auflösung als bisher bringen. Außerdem wurde erst einmal beschlossen, das 68060-Projekt auf Eis zu legen - in Anbetracht der Tatsache, daß Nastas GoldFire so gut wie fertig ist und die Q40 auch schon (wie ich hoffe) in greifbare Nähe kommt, sicherlich keine unvernünftige Entscheidung.

Wie immer - wenn an einem bestimmten Artikel großes Interesse besteht und Sie - lieber Leser - des Englischen nicht so mächtig sind, so ist es kein Problem, den einen oder anderen Artikel aus dem Englischen ins Deutsche zu übersetzen.

■

Endlich ist es wieder soweit! Nicht verpassen! Gleich vormerken!

**Ein großes QL-Treffen für den Raum Österreich,
Schweiz, Süddeutschland, Italien und Kroatien.**

Wo? Im schönen Salzburg!

Ja, ein QL-Treffen in Salzburg wird Ende März steigen. Ansprechpartner ist Herr Koll (zu kontaktieren über Handy 0043 664 160 58 10 wegen Zimmerreservierung). Das Treffen findet statt im Gasthof "Rangierbahnhof" in der Schillinghofstraße 14 (Tel. 0043 662 64 91 89) - es steht ausreichend Platz zur Verfügung. Etwa 14 Personen können in dem Gasthof selbst übernachten, weitere können in der Pension gegenüber untergebracht werden.

Wann? 28. und 29. März 1998!

Anfahrt

Mit dem Auto: Autobahn München-Wien in Salzburg-Nord verlassen und Richtung Rangierbahnhof Salzburg-Gnigl (auf der Straßenkarte leicht zu finden) fahren.

Mit dem Zug: Salzburg-Hauptbahnhof aussteigen, dann eine Station mit Regionalzug fahren, wenige Gehminuten vom der Haltestelle Salzburg-Gnigl bis zum Gasthof. Vom Hauptbahnhof aus sind es auch nur etwa 15 bis 20 Gehminuten.

Mit dem Flugzeug: Shuttle zum Hauptbahnhof und dann wie oben (oder mit dem Fahrrad).

Das Treffen wird offiziell Samstag vormittag starten (Rechner aufbauen, Händler plazieren - und dergleichen), Ende ist offen. Sonntag geht es dann vormittags (ca. 10 Uhr weiter), Ende ist gegen 15 oder 16 Uhr geplant. Es liegen momentan zwar noch keine Details vor, doch ist zu hoffen, daß es wieder (wie so bei den meisten QL-Treffen) nicht nur um Computer und Programme geht, sondern auch noch ein geselliges "Drumherum" stattfinden wird. Wäre also zu überlegen, ob es nicht Sinn macht, schon am Freitag zu kommen und Salzburg zu genießen und einen netten Abend zu verbringen.

Weitere Details zum Treffen wird es im nächsten QL Today geben, daß ich nach Möglichkeit frühzeitiger als notwendig versende, sodaß es jeden noch ein paar Tage vor dem Treffen erreicht.

**Kommt möglichst alle und helft mit, ein weiteres
schönes QL Treffen stattfinden zu lassen!**