

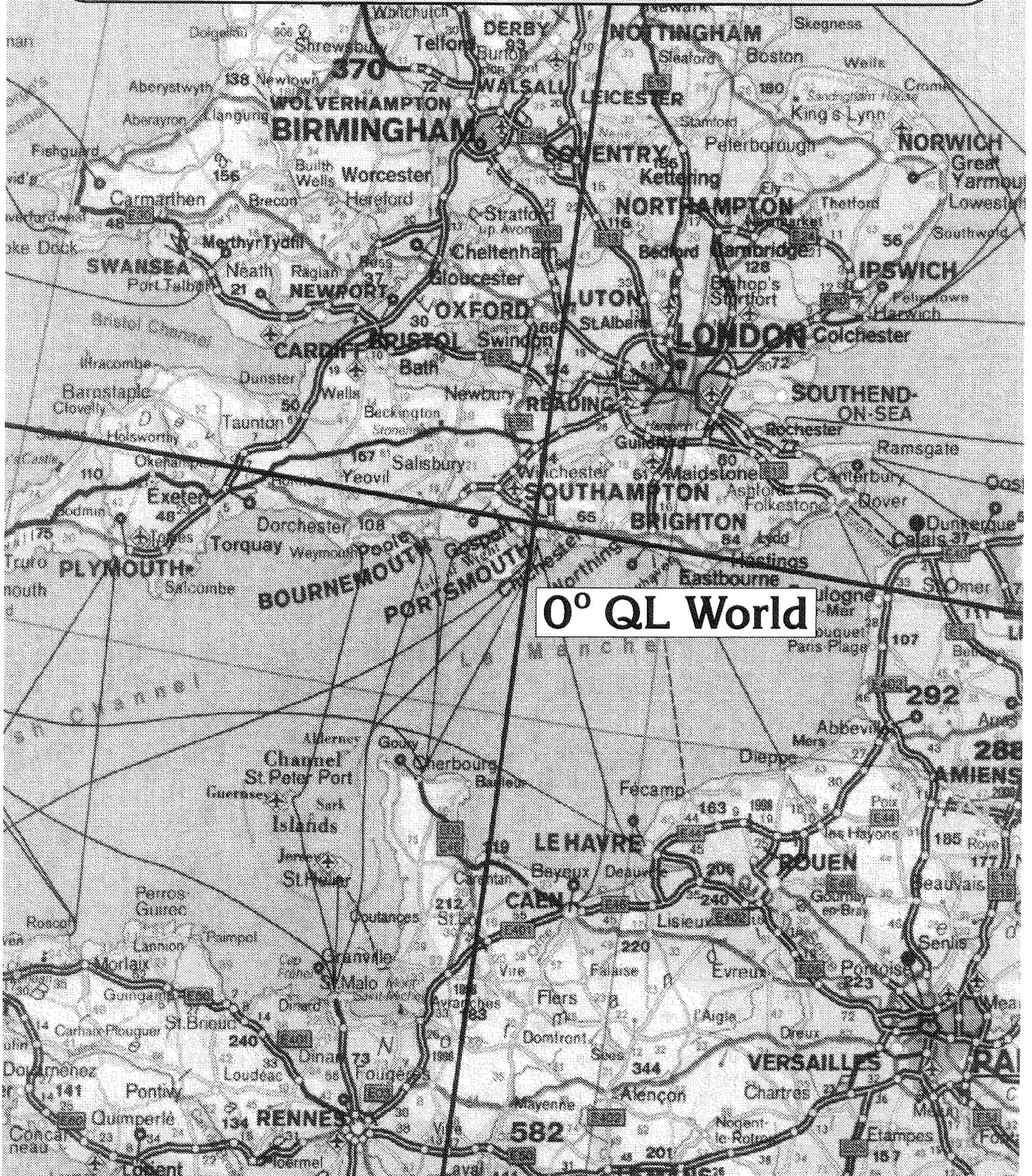
QL Today

DEUTSCH

Jahrgang 5
Ausgabe 3
Sept./Oktober
2000

ISSN 1432-5446

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



0° QL World

Inhalt

- 2 Impressum
- 3 Editorial
- 4 Warum immer ich!? ... oder Glück im Unglück!
Jochen Merz
- 5 Q40 - Eine Zwischenbilanz
Peter Graf
- 7 Zusammenfassung QL Today Volume 5, Issue 2 Juli/August 2000
Wolfgang Uhlig
- 9 Jochen Merz Software
- 11 QPC2 oder QPC2? - Oder wie kann die Zukunft des QLs anders heißen?
Peta Jäger
- 13 Jochens Kommentar
- 14 ProWesS in SBASIC programmieren - warum nicht?
Wolfgang Lenerz
- 17 Kleinanzeigen

Kleinanzeigen

QL Today ist die einzige, aktuelle Zeitschrift für QL-Neuigkeiten. Wir bieten auch Kleinanzeigen zum absoluten Sonderpreis an! 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, verdoppelt sich der Preis.

Aktuelle Kleinanzeigen gibt es in dieser Ausgabe auf der vorletzten Seite - Seite 17!

QL Today DEUTSCH

ISSN 1432-5446

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.de

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. Preisinformationen und Antragsformulare sind bei Jochen Merz Software erhältlich.

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 plziert werden soll, geben Sie es bitte auch an. Natürlich können auch alle Bilder/Artikel auf elektronischem Weg übersandt werden, also in die Box oder per E-Mail.

Redaktionsschluß für Artikel und Werbung:

Ausgabe 1:	30. April
Ausgabe 2:	30. Juni
Ausgabe 3:	30. August
Ausgabe 4:	30. Oktober
Ausgabe 5:	30. Dezember
Ausgabe 6:	28. 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.

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.

Liebe Leser,

ja, Euer lieber Editor war mal wieder vom Pech verfolgt. Dies ist jedoch nicht Grund für die leicht verspätete Auslieferung, denn die war ja geplant und beim letzten Mal angekündigt (im Hinblick auf das QL-2000-Treffen).

Mehr zu meinem Glück im Unglück habe ich in einem Artikel in diesem Heft erläutert. Das Ergebnis ist, daß Q40 bei mir nun endlich seinen Regelbetrieb aufgenommen hat - sehr schön!

Auch habe ich das erste Mal Q40 mit QPC2 zusammengeschlossen. SERNET läuft hier sogar mit 115.200 Baud - toll, das macht richtig Spaß. Bislang war das Maximum bei mir 57.600 Baud und bei den meisten QL-Usern mit SERNET vermutlich 19200 Baud. Von hier ist's natürlich ein massiver Sprung, aber auch die Verdoppelung der Geschwindigkeit bei mir ist überdeutlich spürbar.

So, ich habe auch endlich Nachricht vom Frank Dibowski über die zugesagten und doch fehlenden Artikel erhalten. Frank hatte scheinbar auch etwas Pech und mußte in letzter Zeit zweimal umziehen. Verständlich, daß hier nicht viel Zeit für Artikel und Computerei im Allgemeinen übrig geblieben ist. Er meint aber, zur nächsten Ausgabe die versprochenen Artikel hinzubekommen.

Zum QL-2000-Treffen selbst können wir in dieser Ausgabe logischerweise nichts berichten, sie soll ja erstmalig auf dem QL-2000-Treffen verteilt werden. Mehr dazu sicherlich in der nächsten deutschen Ausgabe.

Nach QL-2000 stehen auch nicht mehr viele Treffen in diesem Jahr an: Paris (aber dafür wird es bei manchen Lesern, die es jetzt das erste Mal lesen, wohl auch zu spät sein) und noch einmal Eindhoven. Wäre schön, wenn wir zum letzten Eindhoven in diesem Jahr mal wieder mehr QLER zusammen bekommen könnten: Es gibt ja unter anderem auch den neuen QPC2 Version 2 zu bewundern (und zu kaufen). Und dann geht's erst wieder im nächsten Jahr mit Treffen weiter... aber das war ja bislang immer so. Die nächsten Treffen stehen voraussichtlich im späten Winter bzw. Frühjahr 2001 an.

Als ganz wichtige Neuigkeit ist zu vermelden, daß Jonathan Dent mittlerweile mit einem QL E-Mils versendet und empfängt. Ja, im Ernst - vor zwei Ausgaben haben wir darüber berichtet (Jonathan machte gute Fortschritte), doch seit einigen Tagen ist es Realität. Hierzu wird QL Today Englisch in dieser Ausgabe sicherlich mehr berichten können, denn Bruce und Dilwyn sind immer in den Newsgroups präsent, in denen heftig darüber diskutiert wird.

Einer von beiden wird eine Zusammenfassung der Situation geben - vielleicht schreibt Jonathan ja auch etwas...? Schauen wir mal, ich kann es noch nicht sagen, da ich mit der deutschen Ausgabe beginne und zum Zeitpunkt des Schreibens dieses Editorials viele Artikel und Neuigkeiten noch nicht hier vorliegen.

Ich kann nur so viel sagen: es tut sich einiges in dieser Hinsicht, und es wird bestimmt nicht mehr lange dauern bis "wir alle" endlich auch mit QDOS und SMSQ E-Mails verschicken können - Anbindung ans Internet vorausgesetzt.

Mehr hoffentlich dann auch übersetzt in der nächsten Ausgabe.

Bis dahin verbleibe ich mit herzlichen Grüßen

Jochen Merz

Warum immer ich!? ... oder Glück im Unglück!

Jochen Merz

In den vergangenen paar Wochen wurde ich hardwaremäßig ziemlich hart gebeutelt: Erst ging der schnelle serielle Port SER4 eines meiner ATARI TTs kaputt (Dauer-Kurzschluß im seriellen Kabel). Ziemlich tragisch für mich, da dies der schnellste Port ist und ich damit meine beiden TTs miteinander mittels SERNET verbunden habe. So habe ich immer die Daten auf beiden Rechnern als Backup - da kann nicht so viel passieren. Die anderen seriellen Ports sind mit Nadeldrucker, Modem etc. belegt und schaffen auch nur 19200 bzw. 38400 Baud. Gleiches am Mailbox-Rechner, hier hängen auch Modems dran.

Damit nicht genug, "starb" mein Haupt-TT einen Tag nach dem Eindhoven-Treffen. Es roch komisch und nichts ging mehr. OK - Netzteil getauscht. Nichts, kein Lebenszeichen. Panik!

Der Haupt-TT ist doch nicht ohne Grund so genannt, schließlich mache ich damit alles (Rechnungen schreiben, programmieren, usw.) Mindestens genauso schlimm: Ich nutze ihn ja auch noch als ATARI, um QL Today darauf zu layouten und zu drucken.

Aber da stand ja noch der Q40. Mangels 4 seriellen Schnittstellen führte er immer noch ein Schattendasein und hatte mich eigentlich nur auf QL-Treffen begleitet. Gut, ein prima Ersatz für den QL/SMSQ-Teil des TT. Die TT-Festplatte hatte es überlebt, aber an den Q40 kann man ja leider kein SCSI anschließen. Doch die Lösung war relativ einfach: Platte an den Mailbox-TT anströmen und per SERNET auf Q40

übertragen. Das Haupt-Problem war, eben nur zwei serielle Schnittstellen am Q40 zu haben. Eine geht schon für die Maus weg, da bleibt nur noch eine. Muß ich halt umstecken, mal den seriellen Drucker, mal SERNET, mal ein Modem zum Fax-Versenden. Nicht optimal, aber was will ich machen.

Schön hingegen ist die enorme Geschwindigkeit des Q40 (aber darüber schrieb ich ja früher schon). So habe ich nun einige Zeit mit Q40 gearbeitet und bin sehr positiv überrascht. Alle meine Programme laufen - viel schneller! Leider kam ich aufgrund des ganzen Ärgers und der damit verlorenen Zeit nicht mehr dazu, mit den Farben zu experimentieren. Ich war froh, daß überhaupt alles läuft.

(Ich sehe gerade George Gwilt hat sich ausführlich mit den Farben beschäftigt und einen tollen Artikel für die englische Ausgabe geschrieben).

Die Lösung für die ATARI-Seite war leider nicht so einfach. Es gibt zwar Calamus für den PC, aber erst einmal mußte ich alle meine Daten vom ATARI zum PC bekommen. Glücklicherweise hatte ich noch ein SCSI-Wechselaufwerk, das sowohl am PC als auch am ATARI funktioniert. Zwar darf nur eine 32MB-Partition darauf sein, sonst "mag" der PC das ATARI-Format nicht, aber besser so als gar nicht. Nachdem ich nach einigen Stunden alle meine Daten "drüben" hatte, fingen die Probleme richtig an. Windows druckt parallel richtig, schließe ich den gleichen Drucker aber seriell an (denn der Parallelport hängt am Q40

und ich will hier nicht auch noch ewig umstecken), dann werden teilweise andere Daten geschickt und der Ausdruck ist nie richtig. Lösung: neue Druckertreiber aus dem Internet. Dann ging gar nichts mehr. Also keine Lösung - Umstecken! Nun ja, auf die ATARI-PC Lösung gehe ich nun nicht weiter ein, sie ist einfach nicht zufriedenstellend.

Weiter mit Q40. Ein wenig umgewöhnen muß man sich schon. Die Bildschirm-Auflösung ist ähnlich wie beim TT und auf einem 17"-Monitor gut zu lesen. 15" muß man schon haben, 14" ist zu klein. Aber selbst 17"-Monitore kosten ja heute nicht mehr viel. Da der TT "nur" 4 Farben kann (also QL Mode 4), der Q40 aber 65536 Farben hat, belegen die gespeicherten Bildschirme viel, viel mehr Speicherplatz als zuvor. Konnte ich in den 20MB meines TT locker 30 Bildschirm füllende QSpreads aufrufen ohne in Speicherplatznot zu gelangen, so "meutert" Q40 schon bei 8. Klar, nimmt sich doch jeder Job gut 1MB nur für's Wegspeichern des Bildschirmhintergrunds. Für die Mailbox-Funktion, für die Q40 eigentlich gedacht war, reichen 16MB ganz und gar - ich würde aber jedem empfehlen, Q40 mit 32MB auszustatten.

Schon nach wenigen Tagen nervte die Umsteckerei unglaublich und so entschied ich mich, eine neue serielle Karte zu suchen. Anfangs, als ich Q40 bekam, hatte ich zwei Karten mit allem Drum und Dran (Harddisk, Floppy, 2 Ser, 1 Par) im Rechner stecken, doch es gab Interrupt-Konflikte (ja, das leidige PC-Problem - es stecken ja auch PC-Karten im Q40) und so mußte ich eine Karte entfernen. Es gab noch frei konfigurierbare ISA-Karten (kein Plug&Play), zwar nicht

preiswert, aber erhältlich. Die Karte hat 4 serielle und 3 parallele Schnittstellen und muß daher zusätzlich zur bisherigen Karte eingesteckt werden, sonst hätte ich ja keinen Floppy- und Harddisk-Anschluß. Kurze Zeit später hatte ich die Maus an allen vier seriellen Schnittstellen laufen - die Freude war groß!

Alle drei PARs liefen auch auf Antrieb. Wow!

Zuviel der Freude, der Rechner wollte auf keinen Fall mit meinem anderen TT mehr über SERNET sprechen. Ich habe konfiguriert was nur möglich war, bis ich dann zum Schluß mit einem Schnittstellentester an SER4 ging und feststellen mußte, daß SER4 an meinem anderen TT total kaputt war - alle Leitungen bis auf Carrier Detect tot. Tja, weshalb weiß

ich bis heute nicht, aber daran ändern kann ich auch nichts. Ob es einfach nur unglücklicher Zufall war oder ob wieder ein Kurzschluß schuld war - keine Ahnung! Gerade serielle Ports sind ja für "hot-plugging" (Ein- und Ausstecken während der Rechner läuft) und sollten auch falsches Einstecken (z.B. nicht gedrehtes Kabel statt gedrehtem Kabel) überleben. Falsches Kabel war jedenfalls nicht die Fehlerquelle hierfür, es war ein richtig gedrehtes Kabel. Naja, die Rechner haben ihr Alter und waren ja auch viele Jahre treu. Dann halt mit Umstecken über einen langsameren Port ... besser als gar nichts. Dachte ich. Dabei blieb es - sobald man Daten schneller als 300 Baud schickt, geht die Hälfte oder mehr verloren. Also wieder am Anfang!? Verzweiflung, viele

Telefonate mit Tony Tebby, probieren, tja, und nun endlich laufen die seriellen Ports. Man muß nur für jeden Port einen anderen Interrupt nehmen und die Karte ist glücklicherweise sehr frei konfigurierbar, so daß ich z.B. 7 und 11 für die seriellen Ports 3 und 4 nehmen kann. Es läuft! Hurra! Noch besser: mein Haupt-TT kam repariert zurück. Nun alles zurück gesteckt (das wird 'ne Zeit dauern), zurück kopiert, dann kann Q40 den Mailbox-TT übernehmen und endlich seine Bestimmung aufnehmen.

Zur Zeit kopiere ich die Mailbox-Daten über einen langsameren Port auf den Haupt-TT, der wird dann nachher abgesteckt und durch Q40 ersetzt. Wie's weiter geht, erzähle ich in der nächsten Ausgabe.

Q40 - Eine Zwischenbilanz

Peter Graf

Der inzwischen schon nicht mehr ganz so neue QL-Nachfolger Q40 hat sich längst zu einem ausgereiften Gesamtsystem entwickelt und sich als eine treibende Kraft beim Fortschritt der QL Soft- und Hardware erwiesen. Vielleicht der richtige Zeitpunkt, um eine Zwischenbilanz zu ziehen. Beginnen wir mal mit einem Rückblick:

Es ist nun schon fast drei Jahre her, daß der Q40 seinen ersten öffentlichen Auftritt hatte. Das war beim QL-Treffen in Eindhoven im Herbst 1997. Ich hatte meinen ersten und einzigen Prototypen dabei. In fieberhafter Arbeit hatte ich die Hardware in Betrieb genommen und eine ganze Menge an Software geschrieben, sozusagen ein Mini-Betriebssystem nebst

Demosoftware, um die Funktionen testen und vorführen zu können. Ich konnte damit eine Festplatte ansprechen und auch Highcolor-Farbbilder anzeigen. Claus hatte sogar noch einen Treiber beige-steuert, der Zeichen mit QL-Fonts am Bildschirm ausgeben konnte. Tony Tebby war bei diesem Treffen anwesend und in eine Menge Diskussionen verwickelt. Ich hatte noch nie mit ihm persönlich gesprochen und wußte auch nicht, ob ihn diese neue Hardware wirklich interessieren würde. Einmal kam er an unserem Tisch vorbeigelaufen und schlug fröhlich mit der Hand auf den Bildschirm, ohne aber irgendetwas zu sagen.

Am Nachmittag kam Tony Tebby dann, um sich die Hardware anzuschauen und sich

über eine Menge Details zu informieren. Ich wußte es damals noch nicht, aber später erfuhr ich, daß sich Tony Tebby innerlich bereits entschlossen hatte, SMSQ/E für den Q40 zu entwickeln. Es sollte zwar noch ziemlich lange dauern, bis er tatsächlich loslegte, aber dieser Tag war ein großer Meilenstein hin zu einem neuen QL.

Die aktuelle QL-Hardware zu diesem Zeitpunkt bestand wahrscheinlich aus einer Aurora-Karte als QL Mainboard-Ersatz, SuperGoldCard, Qplane Backplane, Qubide IDE Interface, sowie diversen Kleinigkeiten wie Chips aus dem Original-QL, Tastaturinterface, usw. Brauchbar, aber es erfordert einiges an Bastel-Arbeit, um ein solches System zu bauen. Und manche der Teile wurden schon gar nicht mehr produziert.

Am Ende waren bei mir immer viele Wunsch offengeblieben. Zum Beispiel, endlich viele

Farben darstellen zu können, und zwar bei hoher Auflösung und mit schnellem Grafikaufbau. Eine schnellere CPU. Mehr Speicher. Schnellere Peripherie. Richtigen Sound. Einen Erweiterungsbus, für den es auch sinnvolle Karten zu kaufen gibt. Ich wollte gerne wieder ein kompaktes und stabiles QL-Gesamtsystem, einen richtigen neuen Computer aus einem Guß.

Die Q40 Hardware war nun da, mit den besten Voraussetzungen, dies alles zu bieten. Es begann die Zeit der Q40 Betriebssystem-Entwicklung, Pionier auf diesem Gebiet war aber nicht Tony Tebby, sondern Mark Swift. Er hat es als erster geschafft, ein Betriebssystem an den Q40 anzupassen, und zwar QDOS Classic. QDOS Classic unterstützt viele Features des Q40, unter anderem IDE Festplatten und Sound. Es bietet hohe QL-Kompatibilität, insbesondere für ältere QL-Software. QDOS Classic ist frei verfügbar, inclusive Quellcodes, z.B. im Internet unter

<http://pages.unisonfree.net/mswift/files/QZ>

Die Q40 Hardware hatte ich mittlerweile zur vollen Serienreife gebracht, und eine erste Kleinserie hatte ich bereits produziert, um Entwickler mit Mainboards zu versorgen. Ein Jahr später war es dann endlich so weit: Im Herbst 1998, wieder in Eindhoven, stellte Tony Tebby die erste Version von SMSQ/E auf dem Q40 vor. Wirklich ein großer Moment, vor allem auch deshalb, weil Tony Tebby den Q40 als seine eigene Haupt-Entwicklungsplattform verwenden wollte. Es war nun auch für die Zauderer und Zweifler unübersehbar, daß der Q40 wahr wird. Qbranch entschloß sich, den Q40 in Serie zu produzieren. Es gab noch einige Verzögerungen,

aber seit eineinhalb Jahren kann jedermann einen Q40 kaufen.

Wirklich erfreulich ist, daß der Q40 eine Rolle als Wegbereiter für einige wichtige Software-Entwicklungen im QL Bereich spielen konnte. Allem voran natürlich die neuen SMSQ/E Farbtreiber. Mit dem Q40 war endlich ein QL-kompatibles 68040 System vorhanden, dessen Grafikfähigkeiten genügend Anreiz geschaffen haben, um diesen Wunsch vieler QL-User noch rechtzeitig zum neuen Jahrtausend zu verwirklichen. Tony Tebby sei es vielenmals gedankt.

Der Q40 war das erste System, auf dem die neuen SMSQ/E Farbtreiber liefen, und zwar mit erstaunlicher Geschwindigkeit. Man muß sich dabei vor Augen halten, daß zum Beispiel beim Q40 im Modus mit 1024x512 Punkten bei 65536 Farben mehr als 30-mal so viele Bildschirmdaten übertragen werden müssen als beim guten alten Mode 4. Es entstanden neue Grafikprogramme, zum Anzeigen von Bildern in verschiedenen Dateiformaten, zum Editieren von bunten Sprites, zum Konvertieren in andere Formate oder Erstellen von Hintergrundbildern. Auch das Erstellen von 3D Grafiken mit dem Raytracingprogramm Povray geht jetzt viel schneller vonstatten, weil der Q40 einen Gleitkomma-Koprozessor besitzt. Ein Luxus, der z.B. auf SuperGoldCard oder QXL noch fehlt.

Auch in anderen Bereichen hat sich einiges getan. So wurde beim Q40 von Tony Tebby erstmalig das QL Sampling Sound System eingeführt, welches die Wiedergabe von gesampelter Sprache und Musik über das Betriebssystem ermöglicht. Und mittlerweile wurden für den Q40 verschiedene Pro-

gramme entwickelt, die diverse Sound-Formate verarbeiten und wiedergeben können. Möglich ist dies, weil der Q40 eine einfache, aber effektive Stereo-Sound Hardware hat, mit Ausgängen für Kopfhörer, Aktivboxen oder Stereo-Anlage. Das Sampling Sound System wurde übrigens auch für QDOS Classic realisiert.

Bei der übrigen Systemsoftware gibt es auch Fortschritte, wie z.B. ein Partitionierungsprogramm, mit dem man direkt unter SMSQ/E eine IDE-Festplatte einrichten und in mehrere logische Laufwerke aufteilen kann. Noch nicht verwirklicht, aber bereits von Tony Tebby in der Dokumentation aufgeführt, sind neue Konzepte, wie z.B. der PROT_MEM Befehl, mit dem man Speicherbereiche vor ungewollten Zugriffen schützen kann. Verwendet wird hierzu die Memory Management Unit (MMU) des Q40, ebenfalls ein Funktionsblock, den es bei früherer QL Hardware noch nicht gab.

Leider ist es so, daß Motorola schon seit längerem die Prozessoren der 680x0 Serie kaum noch weiterentwickelt. Alle hardwaremäßig QL-kompatiblen Systeme können deshalb in punkto Preis und Prozessortakt längst nicht mehr mit Intel basierenden PCs mithalten, auch der Q40. Motorola baut leider keine 68040 CPU, die mit 800 MHz läuft. Wenn man sich aber anschaut, welchen Ärger die Arbeit an einem Intel PC, insbesondere unter Windows bereiten kann, so hat QL-kompatible Hardware noch immer ihre Berechtigung. Ein Q40 kann QDOS oder SMSQ/E innerhalb weniger Sekunden booten, ohne langes "Orgeln" auf der Festplatte, einfach einschalten und los geht's. Und die Arbeitsgeschwindigkeit ist so hoch, daß ich bisher nur sehr

positive Rückmeldungen dazu bekommen habe. Und das, obwohl SMSQ/E in der momentanen Version den Cache des Q40 nicht einmal voll ausnutzt. In früheren Versionen war er aber bereits voll aktiviert, so daß Tony Tebby dies hoffentlich bald wieder beheben wird. Ansonsten kann man auch QDOS Classic verwenden, wenn man, z.B. für Berechnungen, die volle Geschwindigkeit des Q40 ausnutzen will. Dort kann der Benutzer mit BASIC-Kommandos die Konfiguration des Cache etwas detaillierter einstellen.

Die größte Misere der QL Szene, nämlich den Mangel an Software, kann leider auch der Q40 nicht beheben, auch wenn er einige Impulse gebracht hat. Aber der Q40 bietet eine Abhilfe, wenn es keine geeignete QL Software gibt. Und zwar mit Q40-Linux. Dieses Betriebssystem wurde von dem Programmierer-Genie Richard Zidlicky auf den Q40 portiert und wird von ihm ständig weiterentwickelt. Es ist frei verfügbar, näheres siehe z.B. im Internet unter

<http://www.geocities.com/SiliconValley/Bay/2602/q40.html>

Es gibt eine eigene Q40-Linux Distribution, die auf CDROM erhältlich ist, sogar mit grafischem Installationsprogramm. Man kann Q40-Linux direkt aus QDOS Classic oder SMSQ/E heraus starten.

Der Q40 ist damit der erste und einzige Computer im QL Stil, auf dem es auch Linux gibt, und das obendrein noch sehr professionell und stabil. Es läuft eine Unmenge an Software, vom Internetzugang über XWindows und Webbrowser wie Netscape bis hin zu riesengroßen professionellen Grafikprogrammen wie Gimp. Der Q40 kann unter Linux MP3 Musikfiles abspielen, Audio-CDs digital auslesen und vieles mehr. Für reine QL User ohne Unix-Kenntnisse ist Q40-Linux zwar weniger interessant, aber es zeigt, daß mit diesem Computer vieles möglich ist. Und daß man nicht unbedingt einen Intel PC braucht, um Linux zu nutzen.

Der Q40 kann übrigens auch CDROM Lese- und Schreiblaufwerke ansprechen, und man kann mit Hilfe einer Steckkarte neuerdings auch auch Ethernet betreiben! Für diese Geräte gibt es leider noch

keine QDOS oder SMSQ/E Treiber, aber unter Linux läuft alles schon prima.

Soweit die Bilanz dessen, wie sich der Q40 entwickelt hat und was er jetzt schon kann. Als Ausblick in die Zukunft hier nochmals die Bestätigung, daß es mit dem Q60 in nicht allzuferner Zukunft eine noch schnellere Plattform für QDOS und SMSQ/E geben wird. Der Q60 wird eine Motorola 68060 CPU mit 60 MHz haben, vielleicht auch etwas mehr, aber ansonsten eine dem Q40 sehr ähnliche Ausstattung besitzen. Wer nicht das absolut maximale an Rechenleistung benötigt, ist also auch mit einem Q40 gut bedient und muß nicht warten, bis der Q60 in Produktion geht.

Zum Schluß nochmal der Hinweis für diejenigen, die sich per Internet genauer informieren wollen: Die zentrale Webseite für den Q40 lautet

<http://www.q40.de>

Dank Jonathan Dents neuer TCP/IP Software kann man hoffentlich sehr bald auch unter QDOS oder SMSQ/E dorthin surfen...

Zusammenfassung QL Today Volume 5, Issue 2 Juli/August 2000

Wolfgang Uhlig

Neuigkeiten

Paragraph von Francois Lancault

Die kostenlose Version der Textverarbeitung gibt es nun in Version 1.07 und viele kleine Fehler sind beseitigt. Zu bekommen auf den Seiten von PROGS und Thierry Godefroy. Alle registrierten Benutzer haben die Version 2.03 bekommen, die eine erweiterte HTML-Exportfunktion hat.

Dilwyns Webseite

Auf dieser Seite gibt es die neueste Version von Dave Westburys Photon JPEG-Viewer für die

neuen Farbmöglichkeiten des Q40/QXL/QPC. Außerdem eine ganze Menge anderer Programme. Norman Dunbar hat noch weitere andere Programme zu Freeware gemacht (zum Beispiel DJ Toolkit, BASIC Extensions). Reinschauen lohnt sich sicher.

www.soft.net.uk/dj/index.html

www.soft.net.uk/dj/software/other/other.html

Auf der selben Seite und /software/freeware findet sich auch eine Betatest-Version von Dilwyns DOC2HTML Utility, mit dem einfache Quill-docs in HTML-Seiten verwandelt werden.

Der Anfänger-Club Italien
hat eine neue E-Mail-Adresse:
beginners.geo@yahoo.com

G. Plavecs Webseiten-Adresse
hat sich verändert: <http://kuel.ctw.cc/>
Hier sollten auch Information zum Österreich-Treffen zu finden sein.

Die E-Mail-Adressen aller [meine war da nicht!? - Wolfgang] QLer sind in der Quanta-Datenbank. Zu finden unter:
www.quanta.uni.cc

Daniel Baums Webadresse
wurde letztes Mal falsch wiedergegeben: Sie lautet richtig: www.qldesign.com

John E. Juergens
hat herausgefunden, dass QPC2 offensichtlich mit der neuen Version von DirectX 7A schneller läuft. Also nichts wie hin zur Microsoft-Webseite (schauder!) und downloaden das Ding!

The Wall - Bericht über ein Spiel von Norman Dunbar

Der Autor, uns wohl bekannt als derjenige, der seit längerer Zeit die Artikel über Assembler-Programmierung schreibt, hat sich ganz gegen seine Gewohnheit einmal ein Spiel angeschaut: The Wall von Wolfgang Lehnerz. Die "Installation" des Spiels ist supereinfach, die Regeln womöglich noch einfacher und das Spiel spannend, nicht zu leicht und süchtig machend. Kritikpunkte gibt es nicht! Also, Spielefans, kaufen!

UNPICKIN Line Design von Geoff Wicks

Beim Entwerfen seiner Werbung kam Geoff mit dem Problem in Berührung, die fertige Anzeige elektronisch verschicken zu müssen. Da er normalerweise alles mit LineDesign macht, war der mögliche Weg vorgezeichnet. Die Linedesign-Seite als _pic Datei speichern und dann die pic_Datei in eine Prozess-Gas-Chromatograph-Welt übliche Form zu verwandeln. Nach einigem Suchen fand er auf der Webseite von Jonathan Hudson das Programm Unpic bzw. **QUNPIC**, die Version für QLS. Aus dem Wust von Dateien, die - wie üblich bei Jonathan? - aus dem Archiv zum Vorschein kommen, sind eigentlich nur zwei nötig, das Programm selber und die Dokumentation. Man kann damit pics in BMP, TIFF, PNG, Postscript und PCX-Formate umwandeln.

You and Your Programs - Just good Friends?

Du und deine Programme - einfach nur gute Freunde? Teil 9 - Professionelle Pannen
Goeff Wicks beschreibt in dieser Folge, in welchem Maße schlecht geschriebene Programme ihre Benutzer zur Verzweiflung treiben können. Dabei geht es um Probleme wie Kompatibilität mit anderen Programmen bzw. Plattformen, Bequemlichkeit und Logik der Dateneingabe, Fehlerkoorekturmöglichkeiten, Möglichkeiten, später etwas am Programm verändern zu können usw. Seine Schlussfolgerungen sind: Denk immer an den späteren Benutzer, wenn du ein Programm schreibst, seine Verhaltens- und Denkmuster sind mit Sicherheit anders als deine, er könnte eine Behinderung haben, die ihm das Lesen schwerer macht, er muss nicht unbedingt einen Farbmonitor haben und die neueste SMSQ/E Version, usw.

GEE Graphics! (On the QL?)

Part 17

Herb Schaaf referiert noch einmal an seinem Thema vom letzten Mal: Was ist die kürzeste Entfernung zwischen einer Linie im Raum und einem einzelnen Punkt? In dieser Folge gibt es die Ansicht dazu, und zwar von vorn, hinten und von der Seite. Dazu gibt es ein dreiseitiges Listing.

WXQT2

Bericht von Dilwyn Jones

Schock, Horror, QL Today berichtet über ein Window/Linux-Programm! So beginnt Dilwyn seinen Artikel über eines der nützlichsten Programme der gesamten PC-Welt (zumindest für QLer). Die letzte Version des Programmes von Jonathan Hudson (was macht der Mann doch alles für uns!) erlaubt nach dem Installieren auf dem PC unter Windows 95/Linux das einfache Austauschen zwischen Dateien von einem QXL_WIN Dateiensystem zum PC oder umgekehrt. Klingt einfach, aber war doch lange Zeit nicht oder nur über Disketten möglich. Die früheren Versionen dieses Programmes funktionierten noch nicht gut bis hin zum Komplettcrash des PC und anschließender Neuinstallation von Windows, jetzt jedoch scheint alles gut zu gehen.

Man muss sich 3 Ziparchive herunterladen: wxqt2.zip, das ist die Programmoberfläche, qxltool.zip, Version 1.13 oder später, qltools.zip, Version 2.14 oder später. Beim Auspacken überfallen einen wieder eine Unmenge Dateien (hatten wir das nicht schon mal?) und man sollte sich die Dokumentation gut durchlesen um zu

JOHANN MERZ SOFTWARE

Im stillen Winkel 12 D-47169 Duisburg
Tel. 0203 502011 Fax 0203 502012
<http://www.j-m-s.com/smsq/index.htm>

QPC2

Version 2

Ganz neue Version	DM 249,-
Upgrade von QPC2	DM 79,90
Upgrade von QPC1	DM 129,90

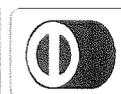
Bei Upgrades bitte in jedem Fall die Original-Disketten mitschicken!

LIEFER- und ZAHLUNGSBEDINGUNGEN

Versandkosten [Deutschland] DM 8,99 (wenn Rechnungsbetrag unter DM 50,- dann nur DM 5,99). Bei Rechnungsbeträgen über DM 500,- kostet es



DM 18,99 [Europa] DM 14,50 (wenn Rechnungsbetrag unter DM 50,- dann nur DM 9,50). Alle Preise inkl. 15% MwSt. Irrtum und Preisänderung vorbehalten. Verrechnung, Eurochecks und Kreditkarten werden akzeptiert. Bankeinzug möglich.



wissen, was man braucht. Aber, wie gesagt, wenn's funktioniert, funktioniert's gut und ist eine erhebliche Erleichterung für Leute, die viele Dateien zwischen den Plattformen austauschen müssen.

Programmieren von ProWesS in SBASIC - Teil 2

von Wolfgang Lehnerz

Im zweiten Artikel seiner Serie steigt Wolfgang Lehnerz nun voll in die Programmierung ein. Auf 5 Seiten wird ausführlich und gründlich erklärt, wie man das macht und anschließend an einem Beispielprogramm erklärt. Leider ist das nicht dazu geeignet, es zusammenzufassen. Es erinnert mich an das Programmieren mit EasyPTR: wenn man die Befehle erstmal verstanden hat, kann man damit auch programmieren. Genau das zeigt uns Wolfgang Lehnerz. Das muss einfach mal übersetzt werden! Findet sich denn niemand? **[Wolfgang macht es selber!]**

BASIC-Linker

ein Bericht von Tim Swenson

Basic Linker ist ein Programmierwerkzeug von Wolfgang Lehnerz (schon wieder er?), das von Jochen Merz verkauft wird. Basic Linker fasst (linkt), kurz gesagt, mehrere, in einem Editor ohne Zeilennummern geschriebene, Basic-Programme zusammen und versieht das neu entstandene Programm mit Zeilennummern. Es kann ein Programm parsen und Syntaxfehler finden, und anschließend Qliberator starten und das Programm kompilieren. Am besten harmoniert BasicLinker mit QD als Editor. Fehler beim Parsen werden nicht nur angezeigt, sondern es wird automatisch in die Zeile gesprungen (in QD), in der der Fehler sich befindet. Langes Suchen ist also nicht mehr nötig. Kleine Nachteile: BasicLinker scheint NICHT SBASIC zu parsen und kann nicht aus QD heraus aufgerufen werden. Ansonsten ein Programm, das nach Tim Swensons Meinung ein gutes Werkzeug für den Basic-Programmierer ist.

Programmieren in Assembler - Teil 8

von Norman Dunbar

Ein Artikel von knapp 9 Seiten. Sicher wieder sehr informativ für Leute die diesem Kurs von Norman Dunbar folgen. Für den weiteren Kommentar dazu siehe die vorhergehenden Ausgaben.

COSMOS

ein Bericht von Brian Kemmett

Schon mal Sterne gesehen? Nein, echt?! Es gab mal ein Programm von Talent Software im Jahre 1985, mit dem man den Sternenhimmel auf dem QL betrachten konnte. Dieses Programm ist von RWAP neu aufgelegt und den modernen Möglichkeiten des QL angepasst und kostet nur ca. DM 15,-. Nach der Eingabe seiner geographischen Situation kann der Benutzer auf dem erscheinenden Sternenhimmel mit dem Cursor verschiedene Objekte ansteuern, ob das nun einzelne Sterne oder Sternbilder sind. Es erscheinen dann die verfügbaren Informationen darüber wie Abstand zur Erde, Name, in welcher Konstellation, Größe, Farbe, Position. Man kann sich auch einen Screendump anfertigen lassen. Brian ist der Meinung, dass dieses schlichte und unpräziöse Programm eine kleine Perle ist und sein Geld mehr als wert.

Inside GoldFire - Teil 1

"Nasta" hat freundlicherweise die gegenwärtigen Spezifikationen seiner GoldFire zur Verfügung gestellt, so dass QL-User sich vorab informieren können über dieses doch immer wieder erwähnte Projekt. Der (bzw. die folgenden) Artikel ist sehr technisch und eigentlich nur für Hardware-Spezialisten wirklich verständlich. Eine Zusammenfassung ist mir (Wolfgang) jedenfalls nicht möglich.

Wolfgang hat es in seiner Zusammenfassung ja angesprochen: Es gibt Artikel, die MÜSSEN einfach übersetzt werden. Also, Hilfe ist willkommen, ein Artikel hier oder ein Artikel da, vielleicht einen oder zwei pro Jahr, ist ja nicht viel. Korrektur-Lesen machen wir auch! Wär' doch schön, wenn sich mal der eine oder andere Leser aufraffen könnte und etwas zu QL Today beisteuern würde. Material gibt's genug! Meldet Euch doch mal!

QPC2 oder QPC?

Oder wie kann die Zukunft des QLs anders heißen?

Von Peta Jäger

Diese Frage klingt zunächst einmal ziemlich blöd. Sie ist aber meines Erachtens momentan für die QL-Gemeinde eigentlich die einzig wahre Antwort auf die Frage, wie die Zukunft des QLs heißen kann.

Q40 ist zwar eine schöne Sache und ich bewundere den Enthusiasmus, der für die Entwicklung aufgebracht wird. Ich habe weiterhin großen Respekt vor der Arbeit, die Tony Tebby in die Verwirklichung eines würdigen QL-Nachfolgers steckt. Auf der anderen Seite bin ich jedoch über die mangelnde Weitsicht Tony Tebbys hinsichtlich einer wirklichen QL-Plattform (in Soft- oder Hardware) über alle Maßen enttäuscht. Zunächst einmal ist es ein Schlag ins Gesicht derjenigen, die den QL seit Jahren unterstützen, neue Hard- (QXL) und Software kaufen, sich auf Versprechen verlassen, was alles noch angeblich kommt (Farbtreiber, Diskettenbetrieb im Hintergrund etc.) wenn dann die Farbtreiber zuerst für Q40 programmiert werden und danach erst für die QXL und den QPC. Auch wenn die Programmierung für Q40 deutlich leichter ist, ist es nicht korrekt, die seit Jahren wartenden QXL-Leute so zu behandeln. Immerhin haben diese die QXL für viel Geld gekauft und die meisten haben dann noch mal für SMSQE bezahlt. Außerdem ist der Farbtreiber für die QXL schon vor Jahren auf einer QXL in Frankreich vorgeführt worden, wenn auch nicht mit großen Erfolg (es lief wohl auf Tonys Rechner, nicht aber auf den dort vorhandenen Rech-

nern). Dass QPC(2) dann allerdings so völlig unbeachtet als letzter bedient wird, haben Marcel Kilgus (der Programmierer) und alle begeisterten QPC-Benutzer wahrlich nicht verdient! Meiner Meinung nach ist QPC die einzige Alternative zum Ur-QL. Das möchte ich im Folgenden erläutern.

Die von Microsoft entwickelten Betriebssysteme (oder was MS als solche bezeichnet) haben sich leider überall auf diesem Planeten (vermutlich auch sonst wo) verbreitet. Wir konnten das nicht aufhalten. Mehrere Dinge sind wichtig für das Überleben eines Betriebssystems, damit es ihm nicht so geht wie OS/2, Atari- und Amiga-OS oder anderen Betriebssystemen wie z.B. unserem QDOS, die nur noch unter "ferner liefen" zu finden sind:

1. Abwärts-Kompatibilität, was Software betrifft.
2. Breite Verfügbarkeit (sollte überall laufen, Notebook, PDAs etc.).
3. Es sollte parallel zu anderen Betriebssystemen laufen.
4. Es sollte alle verfügbare und auch in Zukunft erhältliche Hardware unterstützen.
5. Es sollte nicht teuer sein.
6. Geschwindigkeit, die das Arbeiten angenehm macht.
7. Moderne und einfach zu bedienende Software.

All diese Dinge, die ICH wichtig finde, werden momentan in ihrer Gesamtheit NUR von QPC(2) erfüllt.

Zu 1. Q40 als einzig verfügbare (nicht emulierte) QL-Alternative bietet kein bisschen mehr Kompatibilität als QPC2, eher weniger im Moment. Das ist

zumindest das, was ich gehört habe, denn ich besitze keinen Q40. Allerdings wird die Kompatibilität auch in Zukunft mehr mit dem Betriebssystem zu tun haben, als mit der Plattform. Unter QPC laufen (zum Teil mit Tricks) alle Programme, die auch unter QDOS liefen. Zum Teil habe ich sogar alte Digital-Precision-Programme zum Laufen gebracht. Aber wie gesagt, die Kompatibilität hat hauptsächlich mit SMSQE zu tun und weniger mit dem Emulator oder der Hardwareplattform. Der Druckerport unter QPC wird korrekt angesprochen, die seriellen Schnittstellen funktionieren mit bis zu 115k Baud ohne erkennbare Probleme (u.a. im Mailboxbetrieb getestet). Floppyzugriff funktioniert mit DOS- und QDOS-Disketten, allerdings nicht mit ED-Disketten, die mit 3,2 MB formatiert sind, wie es mit der Goldcard möglich ist. Festplattenzugriff über das QXL-WIN File hat bislang noch nie Probleme gemacht. DOS-Device ist noch in der Erprobung, steht aber auch kurz vor der Veröffentlichung und dürfte eine kleine Revolution darstellen, da man dann endlich vernünftig Daten zwischen DOS und der QXL_WIN Datei austauschen kann. Netzwerk in der klassischen Form funktioniert natürlich nicht, da QPC "nur" das DOS/WINDOWS Netzwerk nutzt. Sind allerdings zwei DOS-Rechner mit installiertem QPC vernetzt, dann kann QPC natürlich auch auf den zweiten PC zugreifen und verfügt damit über das schnellste Netzwerk unter allen QL-Varianten, in der Performance nur abhängig von der installierten PC-Hardware!

Zu 2. Die Verfügbarkeit von Q40 ist wie bei jeder proprietären Hardware naturgemäß sehr schlecht. Selbst so eigensinnige Betriebssysteme wie

z.B. Apples MacOS gibt es für PC-Emulatoren zum Einsatz im Notebook usw. QPC2 läuft praktisch auf jeder Hardwarekonfiguration mit Prozessoren von AMD, Intel oder auch Cyrix mit mindestens 166 MHz. Es läuft in Notebooks, Desktopsystemen und selbst in Emulatoren. Man kann ihn sozusagen überall hin "mitnehmen"!

Zu 3. Q40 läuft zwar parallel zu WINDOWS, allerdings nur wenn man sich noch einen zweiten PC ins Zimmer stellt. Keine wahnsinnig aufregende und vor allem eine teure Variante! QPC läuft unter MSDOS, WINDOWS 3.1, WINDOWS 9x / NT / 2000 momentan zwar nur im Vollbild, allerdings ist ein Fensterbetrieb möglich und wird auch bald kommen.

Zu 4. Bei Q40 ist das auf Grund des Designs wohl kaum machbar. Allenfalls parallel oder seriell angeschlossene Geräte sind möglich. Bei QPC ist momentan zwar auch nicht mehr anschließbar, aber das könnte sich schnell ändern, wie ich weiter unten ausführen werde. In absehbarer Zukunft dürfte sich für QPC(2) eine absolute Hardwareverfügbarkeit realisieren lassen, da alle heutigen PCs mit USB-Ports ausgestattet sind, für die es praktisch sämtliche Peripherie zu kaufen gibt. Unterstützt QPC den USB-Port in der Zukunft, dann wird man praktisch keine weiteren Treiber mehr schreiben müssen, da der USB-Treiber für SMSQE ausreicht, um alle aktuelle Hardware zu betreiben.

Zu 5. Günstig ist eigentlich keiner von beiden. Wenn man jedoch die geringe Verbreitung von SMSQ in Rechnung zieht, kann man mit dem Preis für QPC oder Q40 leben, wobei QPC bei Vorhandensein eines PCs doch deutlich zu bevorzugen ist.

Zu 6. Q40 ist sehr schnell, wird gesagt, und das wird wohl auch stimmen, denn ein 68040 IST sehr schnell. Aber QPC2 ist auch schnell. Zur Zeit werden praktisch keine PCs mehr unter 400 MHz Taktfrequenz angeboten und damit ist QPC fast so schnell wie eine unmodifizierte QXL. Die jetzt verfügbaren und bald standardmäßig erhältlichen PCs kommen mit einer Taktfrequenz von 1000MHz und mehr. Damit wird QPC(2) bald auch Q40 überflügeln. Selbst ein eventueller Q60 wird, abgesehen von den hohen Kosten, nicht sehr viel mehr bieten können, zumal QPC(2) auf aktuelle Hardware (UDMA-Festplatten, AGP-Grafikkarten etc.) zurückgreifen kann. Marcel Kilgus arbeitet momentan an mehreren Projekten, die die Geschwindigkeit von QPC2 bald deutlich steigern werden. Allein der neue Emulationskern wird wohl bereits 20% mehr Leistung bringen. Die Umsetzung der SMSQ/E Fließkomma-Routinen auf den Fließkomma Kern der PC-CPU wird eine Geschwindigkeitssteigerung von rund 400% in diesem Bereich bringen. All diese Dinge sollen sehr bald konkretisiert werden.

Zu 7. Der größte Mangel beim QL ist die Software. Es gibt praktisch nichts. Es gibt auch wenig Entwicklungstools, zumal auch alle Programmierer zum PC abgewandert sind. OK, werdet ihr sagen, wir haben QSPREAD, wir haben QD, Linedesign, SUQCESS, auch diverse Programme von Digital werden wohl noch genutzt, aber ehrlich gesagt, QSPREAD ist doch ein Programm, das nur wenige Leute nutzen, selbst die, die es gekauft haben (ich z.B.). QD ist und bleibt trotz aller Verbesserungen NUR ein Editor, kein Textverarbeitungsprogramm und gerade das ist z.B.

eines der Programme, die beim QL fehlen. Jetzt kommt die Fraktion der Leute, die sagen: "es gibt doch das Programm Paragraph oder Text87 (88,89,90etc.) und Quill." Aber mal ehrlich, Leute, alle diese Programme bieten doch noch nicht einmal die Möglichkeiten, die Wordpad in Windows9x bietet! Zudem ist das auch noch kostenlos beim Betriebssystem mit dabei! Gibt es ein gutes Grafikprogramm (Linedesign ist ein auch von mir geschätztes, jedoch aufgrund der mangelnden Geschwindigkeit und leider mangelnden Weiterentwicklung kaum noch eingesetztes System), das auch aktuelle Drucker unterstützt? Einzig Jonathan Hudson hat im Bereich Datenübertragung den QL auf den aktuellen Stand gebracht, wobei ich mich über seine grafischen Fähigkeiten nicht weiter auslassen möchte. Allerdings geraten wir immer mehr ins Abseits, wenn wir nicht bald Internetanschluss an den Rest der Welt bekommen. Die Programme für E-Mail und Browser sind ja dank Jonathan schon vorhanden, aber statt so etwas (in meinen Augen) Einfaches wie Internetzugang zu programmieren, verschwenden die wenigen fähigen Programmierer in der Szene ihre Kraft für die Anpassung neuer Hardware, die spätestens in zwei Jahren nicht mehr aktuell ist. Die QXL hat es uns doch eigentlich gezeigt, dass man in der heutigen Zeit im Hardwarebereich nicht mehr mithalten kann. QPC geht da den einzig richtigen Weg und sagt: "Hey, wozu Hardware, der PC ist doch leistungsfähig genug. Lasst uns auf Software konzentrieren und Programme entwickeln, die leistungsfähig sind und mit PC-Programmen konkurrieren können." Wer wie ich jeden Tag beruflich mit

PC-Hard und Software zu tun hat, kann ein Lied davon singen, wie schlecht programmiert oder wie unzuverlässig PC-Software ist (QPC ausgenommen :)). QPC könnte da wieder eine Lücke schließen. Mit Superbasic, in C oder Machinensprache kann man nämlich in kürzester Zeit Programme entwickeln, die alles das können, was man so im Alltag wirklich braucht und in aller Regel auch stabil laufen. Wenn Marcel es noch hinkommt, QPC im Fenster laufen zu lassen, könnte man QL-Programme entwickeln, die parallel auf dem Windows Desktop (oder Apple Desktop in der PC Emulation) zu PC-Software laufen. Die Möglichkeiten sind dann unbegrenzt und unser liebgewordenes Be-

triebssystem könnte sich vielleicht sogar wieder etablieren, bzw. zumindest nicht aussterben. Der Q40 hingegen ist und bleibt eine Nische und hat geringe Zukunftsaussichten. Ein Problem ist z.B. - die QXL-Fans können ein Lied davon singen - wenn die Entwickler mal keine Lust mehr auf Q40 haben (z.B. weil die Verkaufszahlen rückläufig sind). Dann ist es für alle, die sich die Hardware gekauft haben und z.B. Ersatzteile benötigen, aus und vorbei! Eine PCI-Grafikkarte, wie sie für den Milan vorgesehen ist, ist jetzt schon im Fachhandel kaum noch zu bekommen. AGP läßt grüßen. ISA-Steckplätze gibt es beim normalen PC auch bald nicht mehr, PC99 Standard! Achtung QXL-Fans: Besorgt Euch

vorsorglich schon mal ein Ersatzboard... P2/2 EDORAMS sind auch immer schwieriger zu beschaffen (Q40).

Für den Bastler und QL-Hardcore Fan mag der Q40 denn auch die einzige Alternative sein, für alle anderen jedoch kann die Wahl eigentlich nur QPC sein!

Für alle Zweifler: QPC läuft bei mir und vielen anderen, die ich kenne, jeden Tag ohne Probleme. Die aktuelle Version 1.54 ermöglicht jetzt den Austausch von ASCII-Daten zwischen Windows-Zwischenablage und Scrap!! Geill!

Zusätzliche Informationen über QPC, Q40 und alles rund um den Sinclair QL auf meiner neu gestalteten Homepage:

<http://www.sinclairql.de>
in bälde...

Jochen's Kommentar

Petas Artikel zwingt natürlich zu Antworten. Ich will nicht auf alle Punkte eingehen, bei manchen hat er auch meines Erachtens nach Recht, doch vergißt er einige Fakten. Vorab möchte ich sagen: QPC ist 'ne tolle Sache, keine Frage, und ich verkaufe sie auch wie jeder weiß. Es ändert jedoch nichts daran, daß QPCs SMSQ/E von Marcel adaptiert wird und Tony nicht die Sourcen daran hat. Das bedingt ganz logischerweise, daß Tony erst die Betriebssystem-Änderungen programmieren muß (auf ATARIs, QXL, Q40 oder was auch immer) und sie dann danach von Marcel angepaßt werden. Da Tony teilweise immens viel ändert, hat Marcel oft einige Zeit damit zu kämpfen.

Zur Implementation der Farbtreiber und "Schlag ins Gesicht". Finde ich schon krass formuliert. Nun ja, wenn ich Tonys Gedanken nachvollziehe

und seine Wahl zwischen einer Maschine mit einer vernünftig funktionierenden Hardware (Q40) oder einem instabilen, langsamen Software-Protokoll (das Tony gezwungenermaßen so machen mußte, weil Stuart von Miracle Systems das QXL-PC Interface nicht unbedingt gut designt hat) und damit bei der QXL unnötige Zeit (ob nun Wochen oder Monate) verschwendet hätte mit Fehlersuche, die mit dem eigentlichen Fehler nichts zu tun haben, dann hätte ich an seiner Stelle sicherlich auch Q40 genommen. Was meinst Du, warum Tony die ganzen Jahre auf einem langsamen Mega STE programmiert hat und nicht auf der viel, viel schnelleren QXL? Da gibt's schon gute Gründe für!

Zudem hat es für die QXL länger gedauert, weil ganz schnell klar war, daß das bisherige QXL-PC-Interface mit der ums Vielfache gestiegenen

Datenmenge für die Bildschirmausgaben in 65536 Farben unbedingt neu designt werden mußte, um es nicht gähnend langsam werden zu lassen.

Bei der ganzen Argumentation wird sicherlich auch vergessen, daß an der QL-Programmierung für Tony sicherlich nichts mehr zu verdienen ist - würde er seine Programmierzeit in richtige Relation zu den Einnahmen stellen, würde gar nichts mehr passieren. Vorwürfe in jeder Hinsicht sind daher relativ unangebracht.

Die Reihenfolge und Zeitdauer ist nicht unbedingt immer wünschenswert, doch auch Tony hat eine Familie zu ernähren, und das geht mit dem QL leider nicht.

Ja, und noch eine Illusion muß ich leider korrigieren: Die Implementation "eines" USB-Treibers soll direkt das Anschließen aller möglichen Peripherie ermöglichen!? Ups, nein, schön wär's, aber jedes USB-Gerät braucht

schon seinen eigenen Treiber. Es mag zwar ein gemeinsames Protokoll geben, doch kann ich einen Drucker nicht wie eine Maus, eine CD-ROM, ein Modem, einen Brenner oder eine Win-TV-Karte ansprechen. Wir hatten dies schon für den Milan 060 diskutiert.

Schauen wir mal, was die Zukunft bringt.

Wie dem auch sei, witzig finde ich die Konstellation, die uns der Zufall beschert hat. Petas Artikel sollte eigentlich in einer der beiden vergangenen Ausgaben von QL Today erscheinen. Es verzögerte sich je-

doch, und ich bat Peter Graf aufgrund der Aktualität der Q40-Situation (auch bei mir), etwas zum Stand der Dinge zu schreiben. Zwei Tage später kommt überraschend Petas Artikel. Besser hätte es kaum passen können.

Hier wurde eine Menge Diskussionsstoff bereitgestellt, und ich hoffe daß es hierauf auch mal kräftig Leserbriefe gibt.

Ich denke auch nicht, daß es nur schwarz/weiß gibt (Q40, QXL) - viele User sind nach wie vor mit ihren QLs, SuperGold-Cards und auch ATARIs zu-

frieden (so auch ich),

Was die provokativen Aussagen zur Software angeht: Mir ist ein Programm mit weniger Fähigkeiten, das nicht crasht, alle mal lieber als ein Word, das nach Belieben meine Arbeit mit "Schutzverletzungen" ruiniert und Fähigkeiten hat, von denen ich 96% gar nicht brauche. Ich nutze seit dem QSpread-Upgrade Qspread mehr als jedes andere Programm

Also - enttäuscht mich nicht und schreibt! Ich erwarte, die nächste halbe Ausgabe von QL Today mit Diskussion zu füllen :-)

ProWess in SBASIC programmieren - warum nicht?

Wolfgang Lernerz

Wenn man sich in den letzten Jahren auch nur ein wenig für den QL interessierte, kann man nicht verfehlt haben, von Prowess zu hören. Es ist mir jedoch aufgefallen, dass viele Leute gar nicht richtig wissen, was Prowess denn ist, und was man damit überhaupt tun kann. Hier ist also eine kleine Erklärung, was Prowess ist, und wie man es in eigenen Programmen benutzen kann:

Zuerst müssen jedoch einige Konzepte erklärt werden, da Prowess am Anfang leicht etwas bange machen kann. Im Gegenteil: es ist ganz einfach! Hierzu kann man sagen, Prowess ist komplex aber nicht kompliziert: Komplex, da man viele neue Sachen erst einmal lernen muss, aber nicht kompliziert, denn das, was man lernt, ist in sich selber nicht schwer zu verstehen.

Einige generelle Gedanken über ProWess

Wie der WMAN der Pointer Umgebung, ist Prowess ein Window Manager. Anders ausgedrückt, macht Prowess nicht anderes als Ihnen zu helfen, Fenster zu erstellen, diese auf den Bildschirm zu werfen, deren Inhalt zu ändern und die Benutzeraktion(en) an Ihr Programm weiter zu leiten. Den Rest des Programms müssen Sie schon noch selbst programmieren! Ob sie WMAN oder Prowess vorziehen ist eigentlich nur eine Frage des persönlichen Geschmacks.

Generell ausgedrückt, ist es Prowess Vorteil über WMAN, dass Prowess Proforma benutzt. Proforma ermöglicht schöne Vektorschriften, d.h. Schriften die (fast) alle Grössen haben können, und immer noch schön aussehen. Der WMAN muss mit den normalen QL-Schriften arbeiten. Vor ein paar Jahren, wurde schon mal darüber gesprochen, Vektorschriften auch für normale QL-Fenster (und also auch für WMAN) einzusetzen, aber dieses Projekt scheint sich im Sande verlaufen zu haben.

WMANs Vorteil über Prowess hat hauptsächlich mit Systemressourcen zu tun, die Prowess und Proforma brauchen: Ohne eine einigermaßen schnelle Maschine, und ohne eine größere Menge an Speicher, sind beide eigentlich nicht zu gebrauchen. Aber das ist auch nur der Preis den man dafür bezahlt, dass man skalierbare Schriften hat, die alle sorgfältig auf den Bildschirm gemalt werden müssen.

Programme die unter Prowess laufen, brauchen immer noch das Pointer Interface selber (aber nicht WMAN, da der von Prowess ersetzt wird). Das Pointer Interface kümmert sich um Sachen, die ein Niveau tiefer liegen, z.B. herausfinden wo der Mauszeiger ist, und was die Maus macht. Der Window Manager, auf der anderen Seite, ist dazu da sicher zu sein, dass Fenster schön auf den Bildschirm gezaubert werden.

Prowess, wie jeder andere Window Manager, ist also nur für den Fensterteil Ihres Programms zuständig. Wenn Sie z.B. ein Programm schreiben, um Dateien von einem Verzeichnis ins andere zu kopieren, dann erstellen Sie bestimmt ein Fenster worin der Benutzer bestimmen kann, um welche Dateien und Verzeichnisse es sich handeln soll. Dieser Teil Ihres Programmes kann von Prowess

unterstützt werden. Der eigentliche Kopiervorgang jedoch ist natürlich unabhängig vom Window Manager, und hat nichts mit Prowess zu tun. Programmieren für Prowess heißt also, Ihre Fenster zu entwerfen und erzeugen. Dies wird dadurch erreicht indem man Prowess "Objekte" erzeugt. Diese können später geändert werden, das Hauptobjekt wird "aktiviert", und wenn die Objekte nicht mehr gebraucht werden, werden sie entfernt.

Objekte

Das Fenster selber ist ein Objekt. Es enthält andere Objekte, die dann auch noch andere Objekte einschließen können, und so weiter. Als Programmierer sprechen Sie nie das Fenster selber an, sondern nur die Objekte dieses Fensters. Normalerweise ist das erste Objekt das erzeugt wird, der "Umriss" des Fensters - alle anderen Objekte werden dann in dieses Umrissobjekt gefügt oder "gegossen".

Der Fensterumriss ist nicht anderes als eine Art Container, dessen Zweck es ist, andere Objekte einzuschließen oder zu "besitzen". Es gibt verschiedene Arten von Objekten, und manche haben sehr viele nützliche Funktionen. Das Umrissobjekt ist der Besitzer aller anderen Objekte die es einschließt. Die Kombination von Umrissobjekt und alles was es enthält wird "**System**" genannt. Ein Programm kann mehrere Systeme besitzen. Sie können z.B. ein System haben, welches dem Hauptfenster entspricht. Danach können Sie noch ein anderes Fenster öffnen ("pull down" wird das für WMAN genannt, und "aktivieren" in Prowess), z.B. um ein Menü zu zeigen. Das wäre dann ein anderes System. Alle Systeme werden jedoch nach einem einheitlichen Modell gebaut: ein Umrissobjekt und dann Objekte die in diesem Umriss enthalten sind.

Anders ausgedrückt, das erste was man beim Programmieren unter Prowess macht, ist ein Umrissobjekt zu erzeugen. Ist dies geschehen, kann man andere Objekte für und in diesem Umriss erzeugen. Der ganze Objektset ist ein System. Wenn Sie so ein System erzeugt haben, können Sie es aktivieren. Aktivieren heißt, dass das Fenster auf dem Bildschirm erscheint, und Prowess kümmert sich dann um die verschiedenen Tastatur- und Mausclicks. Wenn das Fenster auf dem Bildschirm ist, wartet Prowess, dass der Benutzer Objekte (zum Beispiel ein Menüitem) betätigt, oder auf bestimmte Tasten drückt. Diese Information über das Geschehene wird dann an Ihr Programm übergeben. Das erfährt also, dass ein bestimmtes Menüitem ausgeführt wurde, und kann also dementsprechen reagieren.

Objekte (oder Teile davon) ändern

Objekte sind nicht unbedingt statisch, sondern können auch zur Laufzeit noch geändert werden. Nehmen wir als Beispiel ein Objekt, welches eine Zeichenkette enthält, die dem Benutzer einen gerade aktuellen Verzeichnisnamen anzeigt. Jetzt ändert der Benutzer dieses Verzeichnis. Es wäre doch bestimmt nützlich, wenn sich das Objekt, oder zumindest dessen Inhalt, auch ändern würde und das neue Verzeichnis anzeigen. Man könnte natürlich (a) das alte Objekt entfernen, (b) ein neues Objekt erstellen und (c) es dann aktivieren, aber das wäre dann doch viel zu umständlich. Deshalb ist es möglich ein Objekt, oder zumindest viele seiner Aspekte, zu ändern: In dem o.a. Beispiel wäre es möglich, den Inhalt des Objekte (die Zeichenkette mit dem Verzeichnis), und auch andere Aspekte (z.B. die Schriftart) zu ändern.

Objekte anfragen

Es kann auch manchmal nötig sein, dass ein Objekt über sich selbst Auskunft geben muss. Auch wenn Sie ein Objekt erzeugt haben, wissen Sie nicht unbedingt alles darüber (ist ja fast schon Zauberei)! Also können Sie ein Objekt abfragen. So könnten Sie, z.B. ein Objekt haben, welches ein Verzeichnis anzeigt, und dem Benutzer auch erlaubt, dieses Verzeichnis zu ändern. Nachher muss Ihr Programm natürlich wissen was das Verzeichnis denn jetzt ist - Sie können dann das Verzeichnisobjekt anfragen, damit es das jetzt aktuelle Verzeichnis preisgibt!

Objekte entfernen

Wird ein Objekt nicht mehr gebraucht, kann es entfernt werden. Danach existiert es nicht mehr. Interessant ist zu wissen dass, wenn man ein Objekt entfernt, man auch alle von ihm besessenen Objekte entfernt.

ProWesS in SBASIC programmieren

Natürlich können auch Sie Programme schreiben, die Prowess benutzen. Prowess setzt mehr oder weniger voraus, dass Programme die es benutzen in 'C' oder Assembler geschrieben werden, ein Sbasic Interface war am Anfang nicht vorgesehen. Dies gibt es jetzt aber (und wenn Sie in 'C' oder Assembler programmieren, brauchen Sie diesen Artikel wahrscheinlich nicht, Sie sollten sich die (ziemlich gute!) Dokumentation, und die Beispielprogramme einmal anschauen).

Man kann in SBasic also Programme schreiben, die Prowess benutzen, genau wie man auch in Basic Programme schreiben kann, die die Pointerumgebung benutzen. Es ist schon wahr, dass bis

jetzt die einzigen Programme in SBasic die Prowess benutzen, die mitgelieferten Beispielprogramme sind, und auch Agenda (ein bisschen unverschämte Werbung für eins meiner Programme).

Bitte beachten Sie, dass ich "SBASIC" schrieb, und nicht Basic, oder SuperBasic. In der Tat ist es sehr wichtig zu beachten, dass das Sbasic Interface, das ich für Prowess schrieb, nur mit SBasic (also in SMSQ(/E) funktioniert. Mit QLiberator kompilierte Programme funktionieren auf allen Maschinen (die Kompilation wird später noch in einem eigenen Kapitel beschrieben). **Bitte beachten Sie, dass man auf Maschinen, die nicht unter SMSQ(/E) laufen, auf keinen Fall unkompilierte Programme in Basic laufen lassen sollte, die Prowess benutzen. Tun Sie es trotzdem, wird Ihre Maschine früher oder später fast garantiert abstürzen. Es ist klar, dass jedweder Schaden der Ihnen, Ihrer Maschine oder Ihren Daten dabei entsteht, auf Ihre eigene Verantwortung und Kosten geht. ICH HABE SIE GEWARNT!!!!**

(Eine kleine Erklärung für die, die die technische Seite verstehen wollen: Dieser Vorbehalt entsteht nicht durch das Sbasic Interface selber. Um es einfach zu sagen, ruft das Interface nur die von Prowess bereitgestellten Routinen an - und die benutzen Register A6. Sie wissen wahrscheinlich, dass es strengstens verboten ist, Register A6 in einem Programm zu ändern, das unter dem normalen SuperBasic Interpreter läuft. Weder das Sbasic von SMSQ(/E), noch mit QLiberator kompilierte Programme, haben diese Einschränkung. So, wie schreibt man denn jetzt Programme die Prowess benutzen? Ganz einfach mit den neuen Befehlen die das Prowess Sbasic Interface zu Verfügung stellt (es befindet sich normalerweise in einer Datei die PWbasic_rext heisst, die muss man LRESPRn).

Also ist es vielleicht am besten, einfach die neuen Basicbefehle zu besprechen. Ich erkläre natürlich neue Konzepte wenn die auftauchen.

1 - PWcreate: Ein Objekt erzeugen: Besitzer, Typ und "tags"

Alle Objekte werden mit der **PWcreate** Funktion erzeugt. Wie schon oben geschrieben, ist das Erzeugen von Objekte einer der Hauptschritte in der Prowessprogrammierung. Es ist auch meistens einer der ersten Schritte - man kann mit (oder richtiger gesagt: in) Prowess nicht viel anfangen, bevor einige Objekte erzeugt wurden. Das ist also sehr wichtig aber trotzdem leicht zu erzielen, indem man einfach die PWcreate Funktion benutzt. Die hat die folgende Syntaxis:

```
mein_objekt = PWcreate (Besitzer, Typ [{, tag }])
```

Jetzt wird das Objekt erzeugt, und die SBasic-variable **mein_object** enthält (oder richtiger: zeigt auf) dieses Objekt. Von nun an, wann immer Sie ein Objekt für irgendetwas brauchen, können Sie **mein_object** benutzen. Das ist nicht anders, als wenn man eine beliebige Variable benutzt, die von einer Funktion zurückgegeben wird. Manchmal wird auch das Wort "**Object ID**" benutzt, das ist auch einfach nur diese Variable.

Die Parameter, die dieser Funktion übergeben werden, sehen schlimmer aus als sie sind. Wie üblich zeigen die eckigen Klammern '[' einen optionalen Parameter an. Die geschweiften Klammern '{' zeigen einen Parameter an, der beliebig oft wiederholt werden kann, so dass [{tag }] heißt, dass eine beliebige Anzahl von "tags" (was auch immer das sein mag) dem "Typ" Parameter folgen darf, aber nicht muss.

Die PWcreate Funktion erzeugt ein Objekt gemäß den Parametern, die ihr übergeben werden. Jetzt haben Sie ein Prowess Objekt.

Die Parameter der PWcreate Funktion sind logisch strukturiert, und entsprechen einigen wichtigen Konzepten in Prowess.

Der Besitzer Parameter

Der **Besitzer** von mein_objekt ist das Objekt das mein_objekt enthält. Dieser Parameter, also der Besitzer, muss also ein gültiges Prowess-Objekt sein. Wie o.a., gehören Objekte normalerweise anderen Objekten. Mit diesem Parameter sagt man Prowess wer mein_objekt besitzt, d.h. welches Objekt das neu zu gestaltende mein_objekt enthalten soll.

Natürlich kann das zuerst erzeugte Objekt noch keinem anderen Objekt gehören, da es noch kein Objekt gibt das dieses Objekt besitzen könnte, also ist der Besitzer des allerersten Objekts ... 0. Wenn sie nun ein zweites Objekt erzeugen, und geben an, dass mein_objekt davon der Besitzer ist, wird mein_objekt das neue Objekt enthalten.

Der Typ Parameter

Dieser Parameter zu der **PWcreate** Funktion sagt dem Programm welchen Objekttyp sie erzeugen möchten. Prowess hat mehrere Objekttypen, und die haben viele sehr verschiedene Funktionen. Normalerweise würden Sie zuerst ein Umrissobjekt erzeugen, damit Sie den Umriss des Fensters haben. Wie schon gesagt, ist das nur so eine Art Container der alle anderen Objekte des Systems enthält. Hier wäre der Typ des Objekts also Umriss ('outline' in englisch). Ein anders Objekt könnte zum Beispiel ein Menüposten sein, oder ein "infotext" Item.

Jeder Typ hat ein spezielles "Typwort" für sich, das ist vordefiniert und genau im Prowesshandbuch erklärt - aber wir kommen auch darauf später noch einmal zurück.

Wenn Sie also ein Umrissobjekt erzeugen wollen, sollten Sie den Typ "**PW('TYPE_OUTLINE')**" benutzen (was das genau heißt, wird auch noch später erklärt).

Der/die "tag" Parameter

Die verschiedenen Prowessobjekttypen sind sehr unterschiedlich, und ergeben sehr unterschiedliche Resultate. Aber es gibt halt nur eine Handvoll davon, und es scheint deshalb logisch, dass man für jeden Typ noch einige Details definieren muss und kann. Darum kann man auf den Typ-Parameter noch andere Parameter folgen lassen - aus einem unerfindlichen Grund werden diese Parameter 'Tags' genannt. Sie definieren genau die Eigenschaften, die ein bestimmtes Objekt nun haben soll.

Wenn Sie, z.B. ein Objekt des Typs *loose_item* (d.h. einen freien Menüposten) erzeugen, sollten Sie, als Programmierer, schon in der Lage sein zu bestimmen ob das Object eine Ikone oder ein Wort enthält, und wenn es eine Zeichenkette ist, was denn das Wort sein soll. Wenn Sie das Objekt erzeugen, sollten auf den Typ-Parameter also ein oder mehrere Tags folgen, die anzeigen, dass es eine Zeichenkette enthalten soll, und auch was denn das Wort dieser Kette sein soll.

Die Tags sind also immer von dem Typ abhängig. Einer der Hauptarbeiten unter Prowess ist es, zu verstehen, dass jeder Objekttyp seine eigenen Tags hat, auch wenn verschiedene Tags verschiedener Objekte ähnliche Dinge verrichten (z.B. den Text für ein Objekt setzen). Die Beschreibung jedes Typs (im Handbuch) enthält auch die Beschreibung der Tags für diesen Typ.

Tags sind also Parameter für die Typen. In der Einleitung schrieb ich, dass Prowess komplex aber nicht kompliziert ist, es gibt einfach nur ziemlich viel Information, die gelernt werden muss. Der größte Teil dieser Information sind die Tags für die Typen, wie sie heißen und was sie erreichen. Um ganz ehrlich zu sein, kann ich mich nie so ganz erinnern, welcher Typ welche Tags benutzt, und was genau welcher Tag macht. Ich gucke das immer im Handbuch nach und habe das also immer in Reichweite. Aber wenn Sie sie auswendig lernen wollen, habe ich nichts dagegen...

In den meisten Fällen können alle Tags, die man zur Erzeugung eines Objekts gebrauchen kann, auch zum Ändern eines Objekts benutzt werden, auch wenn es da einige Ausnahmen gibt. Auf jeden Fall können alle Tags, die zum Ändern des Objekts gibt, auch bei der Erzeugung benutzt werden.

Ein Beispiel

Hier ist ein Beispiel der **PWcreate** Funktion. Damit wird ein Umrissobjekt erzeugt (das ist der Typ), und das hat ein "Quit" item (ein Tag für diesen Typ). Der Besitzer dieses Objekts ist 0:

```
my_outline = PWcreate (0,PW('TYPE_OUTLINE'),  
PW('OUTLINE_QUIT'))
```

Hier ist der Besitzer 0, es ist also das erste Objekt eines Systems, und es gibt einen Tag für dieses Objekt, nämlich "PW('OUTLINE_QUIT')". Jetzt haben Sie ein Prowessobjekt erzeugt. Es ist ein Umrissobjekt, und dieses Objekt enthält einen (speziellen) freien Menüposten, mit der Zeichenkette "Quit". Wenn sie diesen Umriss aktivieren, erscheint das Fenster auf dem Bildschirm wie durch Zauberei.

Wie man so ein Objekt aktiviert sehen wir in der nächsten Folge dieser Serie.

Kleinanzeigen

Zu Verkaufen

Mega ST4 im Bigtower mit 200W Netzteil - 68030 32Mhz hyperCache Beschleuniger (auf 8Mhz mit 68000 umschaltbar) - 4MB RAM - QVME-Grafikkarte - GE-Soft SCSI Interface - zwei SCSI-Festplatten (200+500MB, bin aber nicht so sicher) - AJAX HD Floppycontroller - Galactic Keyboard - MegaFile Platine ist auch noch eingebaut (erweitert für zwei MFM Festplatten), aber nicht angeschlossen. - eine 1GB IBM SCSI-Festplatte lose dabei - einen Haufen Atari CDs (ein CD-ROM Laufwerk ist aber nicht dabei!) - MagiX! (nicht installiert) - EASE (nicht installiert) - Tandberg SCSI Streamer lose dabei, läßt sich mit GEMAR unter TOS prima verwenden, plus drei Streamerbänder
Alles für nur 100,- DM ... allerdings nur für Selbstaholer, das Ding ist einfach zu schwer zum Verschicken und wäre ja schade, wenn die Post es ruinieren würde.

Ralf Rekoendt, Hennef/Sieg, Tel. 02242-867449 Fax 02242-911834. Ralf.Rekoendt@t-online.de



Zukünftige QL Shows

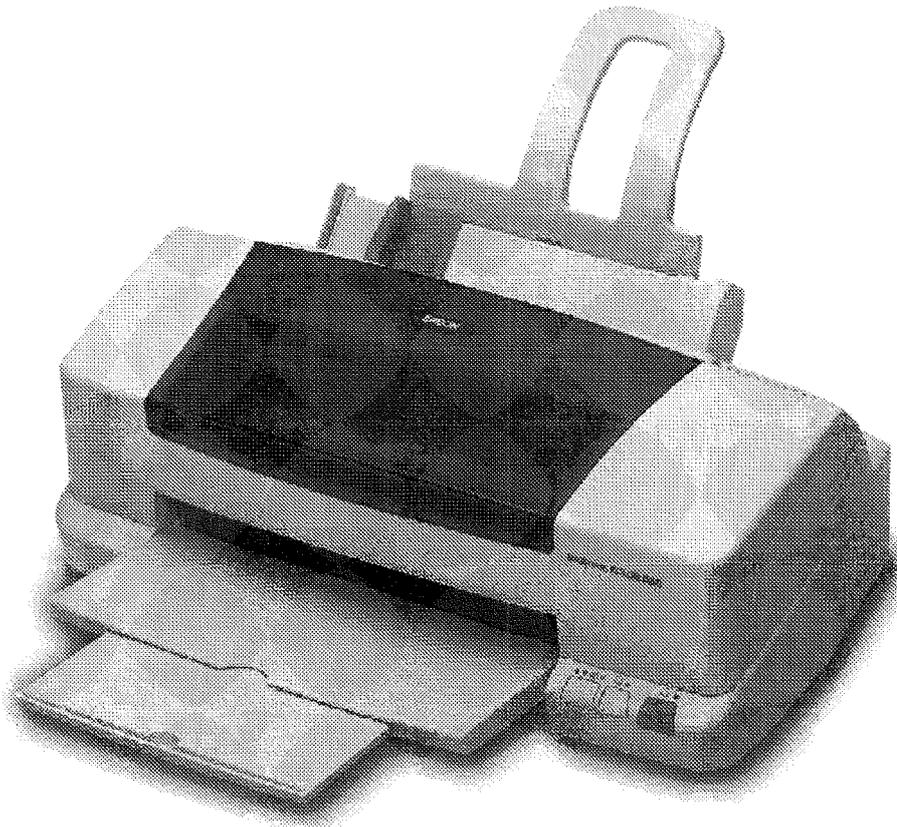


Samstag, 25. November - NL-EINDHOVEN!!!

Die letzte QL-Show in diesem Jahr (zumindest die letzte, von der wir wissen). Natürlich gleicher Ort wie immer - St. Joris College.

Wenn Sie nicht zu QL 2000 gekommen sind (wollen wir nicht hoffen!), oder nach Italien oder Frankreich, dann ist dies nutzen Sie doch diese Chance!

Brandneu: EPSON Stylus Color 880



Schauen Sie mal, was ich gerade noch im Internet gefunden haben! Einen brandneuen Drucker! Mit einer Auflösung von 2880 dpi, Druckgeschwindigkeit von fast 12 Seiten pro Minute (schwarz) und einem empfohlenen Endverkaufspreis von 399,- DM ist das ja wohl der State-of-the-Art Drucker! Und damit auch kein Wunsch unerfüllt bleibt, schreibt EPSON, daß der Drucker ESC/P2 kann - damit auch "QL"-fähig ist. Was mich ein wenig verwundert ist, daß unter "Schriftarten" zwei Bitmap-Schriften und eine skalierbare Schrift steht. Dies ist deutlich mehr als "nur" Raster-Drucker bieten (dort gibt's nur eine Bitmap-Schrift), aber doch weniger als die bisherigen ESC/P2-Drucker können (11 oder 13 Bitmap und 4 skalierbare). Wie dem auch sei, die Daten sind super und man wird sicherlich auch Quill und andere Programme ohne Änderung der Druckertreiber nutzen können. Bei dem Preis kann man ja kaum widerstehen - wer der Versuchung erliegt, kann ja auch ein paar Zeilen an QL Today schreiben.

An alle Leser, Show-Veranstalter & Autoren!

Bitte informiert uns immer möglichst bald über zukünftige QL-Veranstaltungen im nächsten Jahr. Nur so können wir unsere Leser rechtzeitig informieren - und eventuell Einladungs-Mails bündeln. "Sparen" heißt die Devise - und jedes Bißchen hilft!

Eine große Bitte wäre auch das rechtzeitige Einsenden von Artikeln und Beiträgen. Nur so kann QL Today auch in Zukunft rechtzeitig und in der gewohnten Qualität erscheinen - Danke!