

BARCODE

REVIEW

**READ BARCODES
FOR FREE WITH
YOUR QL!**

**PLUS: ALL THE
USUAL NEWS,
LETTERS AND
REVIEWS.**

**YOUR FREE
BARCODE**



Issue 3
2004
FREE

5 12647 86493 3



CONTENTS

July is officially a “blue moon” month (or was that last month? I forget), which might explain why Barcode Review Issue, whatever issue it is, has made an appearance.

Don't be fooled by the new design, the content is as bad as ever, and the design itself was knocked up by two monkeys banging a typewriter (f'nar - Ed). Oh, there are also no previews, reviews or celebrity interviews this month, mostly because I couldn't be bothered (it has taken long enough to write the content that is in this issue, and most of that was cut'n'pasted from email).

So, without further ado, and whilst still trying to think of how best to fill the rest of the contents page, let's get on with - whatever issue it is - of Barcode Review.

(How about a picture of the now-legendary Ethernet T-Piece dog? - Ed) Good idea.



You can make your own little T-Piece critter with six Ethernet T-pieces and a bent paperclip for the tail.

CONTENTS..... 2
You're reading them!

WIBBLE..... 3
Barcode Review talks to the trees (literally) about new RFID technology and the future of barcodes.

LETTERS..... 4
The page that we hand over to you, our lovely readers.

FEATURE..... 5
Find out how to do barcode scanning on your QL.

NEXT MONTH..... 8
And if you've read all that, don't be deceived by the “next month” page, as Barcode Review has not and never will stick to any kind of schedule, monthly or otherwise.

CREDITS

Editor and anything else that doesn't have a credit Chris Young
Contributors Simon N Goodwin

An Unsatisfactory Software publication
<http://www.unsatisfactorysoftware.co.uk>

Write to us at
barcodereview@unsatisfactorysoftware.co.uk

© 2004, may not be sold for profit.
Created entirely with Pagestream on Amiga.



WIBBLE

Barcodes are set to be replaced with RFID devices - little radio transmitters with all the cuteness of... something not very cute at all. This issue, Wibble takes to the streets and asks some inanimate objects what they think

A TREE

WIBBLE: So, what do you think of these RFID things, then?

TREE: Well, it's good, isn't it?

WIBBLE: Good?

TREE: Yes, barcodes are printed on paper, which is made of wood, which means the more of the things that are produced, the more of my cousins are chopped tragically killed. RFID technology doesn't use paper, so us trees can live on in peace.



WIBBLE: (silence)

TREE: Which is good for the environment.



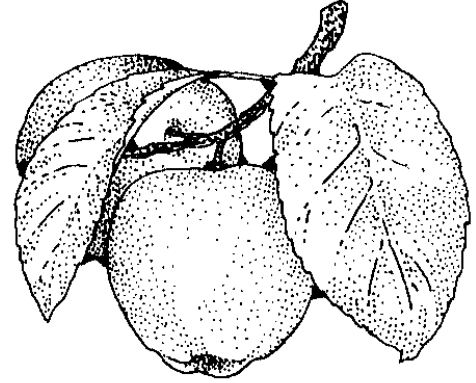
IN THE BINS

The barcode found in the bin this issue was this faded and biological waste encrusted little number. Yeurgh!

Send your disgusting barcodes to the usual address.

AN APPLE

WIBBLE: So, surely you don't also believe these evil electronic tags are good and our lovely, sweet, innocent and friendly barcodes are bad?



APPLE: Well, it doesn't really affect us apples.

WIBBLE: Why not?

APPLE: We don't tend to get barcodes stuck to us, just bagged up and weighed at the checkout.

WIBBLE: Oh.

CELEBRITY BARCODES COMPETITION

YES, FINALLY YOUR CHANCE TO WIN A BARCODE ENDORSED BY A CELEBRITY!

Barcode Review has exclusively obtained ten barcodes from wok boxes, signed by Ken Hom!

All you have to do, is cook a chinese meal that looks like a barcode, and send a photograph of the results to us.

The finest ten entries received before the closing date of the day before this issue of Barcode Review was released, will win this piece of barcode history.



LETTERS

had a whole bag full of barcode-related trivia from around the world ready for the letters page this issue. Unfortunately the dog ate it, so this array of rubbish that was sent to me several years ago will have to do. Send letters for the next issue to barcodereview@unsatisfactorysoftware.co.uk

I WANT A FREE BARCODE BATTLER

could i have a barcode battler for free?ive been searching for one in mint condition for so long....could it be possible?

Killer Chicken, email

No.

Ed

GWOAA (snip - Ed) NN

Gwoaaaaaaaannnnnnnn please print this letter then i'll be the envy of the school playing field!!!!!!!!!!!!!!

go1gwyaa00, email

I don't know why the school playing field will be envious of you getting a letter printed in an Internet magazine which only gets "published" once in a blue moon. I suppose it might go green with envy but, erm, I doubt anybody would notice.

Ed

HAIKU CORNER

Dear Sir, I wish to congratulate you on your most excellent publication. "Barcode Review" is like a breath of fresh air into the normally staid and predictable world of barcodes, and your magazine stands head and shoulders above such rival publications as "Which Barcode?" and the contemptible "Barcode User" (spit, curse etc).

I was greatly moved by Mike T's haiku in issue 2 and felt compelled to add my own to your esteemed magazine:

Mark of the devil
On ev'ry thing you purchase
Argh! run away, shopper!

Hajo Spuunup, email

You were doing so well, up until the haiku started to kick in. I was going to award you the

star letter but I'm not going to now.

Ed

KEEN ON BARCODES

star letter

Roll on issue 3. And perhaps you can mention the program that writes readable and useful barcodes on a ZX printer (really) - I know it was before Your Spinclair time (in the halcyon days of Your Spectrum, the Sinclair magazine that took more than a month to read) but see: http://www.users.globalnet.co.uk/~jimg/yr04/yr04_29.htm

I suspect anyone strange enough to enjoy your mags might like this too. Any if you think I'm keen on barcodes (er, not especially, just keen I guess) bear in mind that my brother Sam worked the Casio format out by inspection, then wrote the program to generate such barcodes on the ZX printer on a 48K Spectrum with microdrives but some keys broken, so if he needed some keywords or glyphs he had to:

- (1) MERGE a program containing the symbol required.
- (2) Find and edit the line with it in.
- (3) Type the rest of the line round the symbol
- (4) Delete the rest of the MERGE'd stuff.

He can also read ASR33 paper tape by eye, but anyone can do that, can't they?

Simon N Goodwin, email

P.S. Must get back to my 1.77 MHz Video Genie now.

Thanks Simon, that fitted in quite nicely here and means I don't have to give that haiku-wielding maniac the Star Letter. Not that there are any prizes associated with the Star Letter, it's more of an ego thing.

Ed

P.S. Must get back to my tea and biccies now.



FEATURE

QCat - really free bar code scanning for Qdos!

Prompted by Tim Swenson and Tony Firshman, I have written a SuperBASIC driver for CueCat barcode scanners which are being given away for free by Radio Shack, the US chainstore known as Tandy in Europe. They are free because Radio Shack want PC owners to use them to scan bar codes from their catalogue, to simplify online ordering. However they work just as well for other purposes, as Linux hackers soon found out, and Qdos ones too!

These gadgets can be persuaded to work with original QL systems and Qdos emulators, and I expect with SMS. I wrote and tested my software in UQLX, the free Qdos emulator for Linux, and Timothy Swenson has successfully read messages from the CueCat into a QL equipped with a Falkenberg Keyboard-90 interface.

A CueCat is a bar-code reader that daisy-chains onto an IBM standard PS/2 keyboard port. You plug a lead from it into the PS/2 keyboard socket, and plug the keyboard into that. The CueCat lights up but does nothing till one of many standard barcode formats passes in front of it, at which point it spews out several

dozen key codes, as if they'd been typed.

Codes is the word; the message consists of raw key up and key down messages, prefixed by the PC code for ALT-F10, followed by encoded data using four six bit codes for each three eight bit bytes, interspersed with other padding. The first 18 bytes you decode are the serial number of the pen, which doesn't change. After that comes a three-character abbreviation indicating the type of bar-code scanned, and the bar code value. Usually this just consists of digits but it's potentially arbitrary text or even binary data.

Armed with various notes and programs from the Linux community it did not take me long to cobble together a SuperBASIC decoder.

It appears that CueCat is not compatible with SuperHermes. But this does not mean that it's incompatible with Qdos. If you connect one to SuperHermes, keyboard input from the PC keyboard collapses as soon as the CueCat tries to insert its gibberish into the input stream.

Normal input can be restored, at least until the CueCat chirps up again, by unplugging both keyboard and CuCat (i.e. unplug the PS2 keyboard adapter from the Hermes, to reset both CueCat and the keyboard). Just unplugging the keyboard and leaving the CueCat connected

does not restore keyboard operation.

Apparently CueCat does not work with the keyboard port on the Q40/Q60 either, for hardware reasons. It appears possible to adapt it for use with Hermes, Q40/60 and other Qdos compatibles with one cheap chip, diverting its messages into a serial port, but I have not tried that; there are details - including a schematic, parts list and construction photos - in the cuecat-0.1.9.tar.gz archive on the web-page I mention at the end of this report.

I inserted the CueCat I received into the PS2 keyboard circuit for my Linux box running Debian 2.4 and managed to swipe a few ISBN codes, starting cautiously with barcodes from books published in the USA. Since then I've scanned magazines and many other types of product code into my Minerva Qdos emulation.

At first this swiping stuffed gibberish into the Linux keyboard queue, and hence into window #0 in SuperBASIC. The CueCat output is basically alphanumeric, plus "+" and "-" to make up the full 64 code set for six-bit values, plus an initial ALT-F10, a few full stops between sections, and a final line-end marker

I opted for stripping out the full stops and some bytes from start and end of the input, and processing whatever was left. My first version converted the data into binary, which was



simple but rather slow. Later I condensed the program, eliminating the need for the binary string and a slew of SElect lines. The first program, CueCat02_bas is the easiest to read, but the second, CueCat04_bas, is shorter and faster. In case you wonder what you may be missing out on, CueCat01 was a prototype that failed to decode some barcodes, and CueCat03 was a hybrid used to check that the new code did the same thing as the binary version.

Anyhow, I fired up UQLX and verified that it could read input redirected from the CueCat. By the time it reached Qdos the message had three characters on the front. These were 255, 250, 46, which starts correctly for ALT and Shift F5 (aka F10) on Qdos, followed by a full stop "." that seems spurious.

Stripping those four off and unpacking the lazy way with BIN and BIN\$ (in the Quanta library and PD Toolkit as well as TK2, so viable for free code development) I was able to extract a plausible CueCat ID from the start of the stuff it inserted: I get '000 000 001 531 875 101' which is not quite that listed in the Linux cuecat kernel patch 0.1.9 document by Pierre-Phillipe Coupard but was close enough to inspire confidence.

I used the recently-enhanced DIY Toolkit function INBYTE\$ to read the data, as I made a version with a convenient timeout parameter as part of my work on the Kodak digital camera driver, with help from

Turbo maintainer George Gwilt. There remains a risk that INBYTE\$ will read only the start of the message if it arrives just as the timeout expires, but the timeout is adjustable and it is convenient to collect all the characters in one call falling out nothing arrives.

Otherwise you could use INPUT or Turbo Toolkit INPUT\$ (without the third timeout parameter) which wait indefinitely for a line, or a loop containing INKEY\$ calls to do much the same thing with more code. All you want is to read a line of input from the keyboard. The CueCat conveniently sends an Enter character at the end of the stuffed string.

A bit more piddling around in SuperBASIC got me code that could read ISBN codes with the five-digit suffix (that usually seems to be 90000 for books that I own) which CueCat labels IB5, and the ISSN number from the front of Amiga Active magazine (categorised as IS2).

It also read the IB5 barcode from the book 'Macintosh Toolbox Essentials'. I had to add a couple of extra lines to get it to support 'short' bar codes such as the bars on the back of the Supercharge box (and many other products) that return a base64 sequence that is a couple of digits short in the last 24 bit packet. This is the reason for the 'short' test in the listing.

These EAN-13 (or E13 in CueCat terms) barcodes are the most common in general retail. After about 20 tries I

managed to read the barcode from the foil around a C90 Goodmans chrome cassette. CueCat doesn't cope very well with small or glossy codes, but I guess that doesn't matter to Tandy/Radio Shack. Another EAN-13 that came up fine was from the Amiga game UFO Enemy Unknown, as repackaged for Microprose/Acid Software by Guildhall.

But it won't read the barcodes on some other software in my collection, including TurboText and Elite for the Spectrum (about the only thing that used Lenslok other than Supercharge). I suspect the problem with those is an interaction between the pen and the old boxes, rather than the barcode data formats.

The UPC-A code on my old Caldera Open Linux Box set was read OK, and got classified as UPA by CueCat. Sinclair's QL Classic Adventures box barcode scanned fine, as E13. So did the barcode on Tascopy QL, which turns out to be an EAN book code which might have saved Tasman some VAT.

Programs
The longer listing, CueCat02_bas, decodes the entire base64 part of the message from the CueCat, but only prints the type of barcode and the barcode itself. The serial number of the pen is in scan\$(1 TO 18) after each successful scan. The second listing skips that invariant prefix, starting to decode from the 27th byte of the message



from the pen, where the barcode type is encoded.

CueCat uses a custom six-bit code to ensure that any message can be encoded into groups of just 64 printable characters. Each clump of four characters encodes three arbitrary data bytes - 4 times 6 bits in yield 3 times 8 bits out. To make life a bit more fun for hackers, the output bytes need to have three bits reversed to yield the correct code. I used the exclusive-OR operation ^ 67 to toggle bits 0, 1 and 6 in SuperBASIC.

CueCat02_bas assembles a 24 character binary string from four bytes, using a SElect structure to find a six bit pattern for each input byte and BIN\$ to convert that to six binary characters, then three calls to BIN to extract eight bit values. CueCat04 skips the binary stage by building a 24 bit value in the variable SUM and splitting that up into three parts in the RECODE\$ function, by division, multiplication, subtraction and INT operations. The DIV and && operators do this sort of thing faster, especially in compiled code, but those only work with 16 bit integers.

The DECODE64 function in CueCat04 replaces a dozen simple lines in CueCat02's DECODE1 with one complicated one. It uses INSTR to find a character in a string. It turned out to be easiest to do this if the alphabet and digits appeared in reverse order, because of the way INSTR is case-

insensitive and CueCat uses lower and then upper case letters for the first 52 encodings; the (CODE(c\$)>"96")*26 part takes away 26 from the result if the character is lower case, differentiating its code from that for the equivalent capital letter. I typed the alphabet backwards by interspersing left arrow presses while typing it in the normal order, as I can remember it easily from A to Z but not the other way round. DECODE64 does not give an error message if the character is not in the expected set, but you could add this by checking for the value 64 before returning - valid codes are 0 to 63, and 64 means the character C\$ was not found in the pattern.

Both programs work consistently here, and CueCat02_bas in particular should be easy for you to read or modify if necessary. I enjoyed the hack and am grateful to Tim and Tony for setting me up for it.. US QI-ers should consider a trip to Radio Shack, especially if they have a PS2 keyboard on their QL-clone.

(I was told to add some jokes to this, but I thought the biggest joke would be if people read it all the way through expecting some jokes **Ed**)

Simon N Goodwin

REFERENCE: CueCat hacking tips, programs and links online:
<http://www.beau.lib.la.us/~jmorris/linux/cuecat/>
- simon@studio.woden.com AKA simon@studio.co.uk

CueCat02_bas

```
100 REMark CueCat
scanner for QDOS/UQLX
110 REMark Simon
Goodwin May 2002, v0:2
120 REMark Uses P.D.
BIN, BIN$, INBYTE$
130 REPEAT wait
140
k$=INBYTE$(#0,80,250)
150 IF k$>" " : EXIT
wait
160 PRINT #0;"?";
170 END REPEAT wait
180 REPEAT l: k="."
INSTR k$ :IF k=0 :EXIT
l :ELSE k$=k$(1 TO k-
1) & k$(k+1 TO)
185 short=(LEN(k$)-3)
MOD 4
187 IF short THEN
k$=k$(1 TO LEN(k$)-1)
& "aaa"
190 scan$=""
200 FOR group=3 TO
LEN(k$)-1 STEP 4
210 sum$=""
220 FOR c=0 TO 3
235 sum$=sum$ &
BIN$(DECODE1(k$(group+c
)),6)
240 END FOR c
250 scan$=scan$ &
DECODEBIN$(sum$)
260 END FOR group
270 PRINT "Barcode
type ";scan$(19 TO 21)
280 PRINT "Value
";scan$(22 TO
LEN(scan$)-short)
370 :
380 DEFine FuNction
DECODEBIN$(t$)
390 LOCAL r$(3)
```



```

400 r$=CHR$(BIN(t$(1 TO 8)) ^^ 67)
410 r$=r$ & CHR$(BIN(t$(9 TO 16))
^^ 67)
420 RETURN r$ & CHR$(BIN(t$(17 TO
24)) ^^ 67)
430 END DEFINE DECODEBIN$
440 :
500 DEFINE FUNCTION DECODE1(c$)
510 LOCAL c
520 c=CODE(c$)
530 SELECT ON c
540   =CODE("a") TO CODE("z")
550   RETURN c-CODE("a")
560   =CODE("A") TO CODE("Z")
570   RETURN 26+c-CODE("A")
580   =CODE("0") TO CODE("9")
590   RETURN 52+c-CODE("0")
610   =CODE("+") : RETURN 62
620   =CODE("-") : RETURN 63
630   =REMAINDER
640   PRINT #0;"Character
"!c$!"' CHR$("!c!") not in CueCat
set"
650   STOP
660 END SELECT
670 END DEFINE DECODE1

```

NEXT MONTH

- News about barcodes
- Previews of forthcoming barcodes
- Reviews of newly launched barcodes
- Features about barcodes

And more stuff about barcodes than you can shake a cat at, in the biggest free on-line barcode magazine with the name "Barcode Review"

Out at an undetermined point in time

CueCat04_bas

```

100 REMark CueCat scanner for
QDOS/UQLX - Uses P.D. INBYTE$
110 REMark Simon Goodwin May 2002,
Streamlined version 0:4
120 REPEAT wait:
k$=INBYTE$(#0,99,250): IF
LEN(k$)>50 THEN EXIT wait
130 REPEAT z: k="." INSTR k$: IF
k=0: EXIT z: ELSE k$=k$(1 TO k-
1)&k$(k+1 TO)
140 short=(LEN(k$)-3) MOD 4
150 IF short THEN k$=k$(1 TO
LEN(k$)-1) & "aaa"
160 scan$="": FOR group=27 TO
LEN(k$)-1 STEP 4
170   sum=0: FOR c=0 TO 3:
sum=sum*64+DECODE64(k$(group+c))
180   scan$ = scan$ & RECODE$(sum)
190 END FOR group
200 PRINT "Type"!scan$(1 TO
3)!"Value"!scan$(4 TO LEN(scan$)-
short)
210 :
220 DEFINE FUNCTION DECODE64(c$)
230 RETURN 64-(c$ INSTR"-
+9876543210ZYXWVUTSRQPONMLKJIHGFEDC
BA")-(CODE(c$)>"96")*26
240 END DEFINE DECODE64
250 :
260 DEFINE FUNCTION RECODE$(t)
270 LOCAL t1,t2,t3
280 t1=INT(t/65536) : t3=t-
t1*65536
290 t2=INT(t3/256) : t3=t3-t2*256
300 RETURN CHR$(t1 ^^ 67) &
CHR$(t2 ^^ 67) & CHR$(t3 ^^ 67)
310 END DEFINE RECODE$

```

