Subject: A simple RS232 bootstrap program for the VIC20
Posted by Chris Baird on Sun, 15 Jun 2014 15:33:31 GMT
View Forum Message <> Reply to Message

I've just tidied up a useful little program of mine that's used over the past year to transfer games and other code to my datasette-only VIC20. It wouldn't be difficult to modify for the '64.

http://kildall.apana.org.au/~cjb/vic20/vic20-rs232-bootstrap .tar

README:

Here's a quick and dirty serial bootstrapping system I've been using with my VIC20.

Cooking instructions:

Firstly, you'll need to provide a serial data connection between your Python-loving system and the VIC-- I'm using a USB TTL Serial adaptor borrowed from an embedded programming setup. (Connect the GNDs, TxD to User port pins B&C, RxD to pin M.)

Secondly, you'll need to type in the BASIC bootstrap problem included below. Stop whining, it's only 25 lines.

In "vicload", edit the serial port device to suit, and then run it like this:

```
$ ./vicload start_address_decimal memory_image
```

or to have the first two bytes of the file used as a start address:

```
$ ./vicload 0 program.prg
```

The packet format is deliberately very simple, to make the receiving BASIC program run fast by the use of INPUT#, which doesn't take to binary data very well.

I've also written a machine-code version of the receiver. "downloader.prg" runs from regular program RAM for downloading cartridges (takes about 4 minutes) and programming projects into BLK5 RAM, and "downloader.a0" that resides in BLK5 for downloading the other way around. The assembler code needs the cc65 tools to build.

(Still included in the source is an attempt to make the downloader
save the transfered data to tape-- unfortunately I discovered a
limitation in the VIC's ROMs when trying to save memory over location
32768, preventing saving binary cartridge images. If anyone knows of a
way around it, let me know!)

V2 BASIC bootstrap

```
100 OPEN2,2,3,CHR$(8)+CHR$(0)
122 DIMB(90)
130 PRINT"<BLK><CLS>";
150 L$=""
160 PRINT"<CLS><BLU>RECEIVING LINE<YEL>";LI
161 PRINT"<RED>----------<BLK>";
170 PRINT#2,LI
180 POKE36879,25
230 INPUT#2,L$
240 PRINTL$
300 ER=0
310 S=VAL(LEFT$(L$,5))
311 IFS=65535THENSTOP
315 IFS<40960ORS>49151THENER=1:S=32768:REM REMOVE IF SO INCLINED
320 P=0:FORI=1TOLEN(L$)-5STEP3
330 A=VAL(MID$(L$,5+I,3))
340 B(P)=A:P=P+1
345 IFP=90THENER=1:P=0
350 NEXT
360 CH=0:FORI=0TOP-2:CH=CH+B(I):NEXT:CH=CHAND255
370 IFB(I)<>CHTHENER=1
390 IFER=1THENPOKE36879,42:GOTO150
500 FORI=0TOP-1:POKES+I,B(I):NEXT
505 POKE36879,93
510 LI=LI+1:GOTO150
```

The same code could be used for the CBM64 by changing/omitting the POKEs

---

--
Chris Baird,, <cjb@brushtail.apana.org.au>
Dedicated to Charl and other Voodoo Castle players...