HC-2000 serial communication experiments

Stefan V. Pantazi

svpantazi@gmail.com

Jun 30, 2023, update 4

Jul 19, 2023, update 5

HC2000 – PC Serial Cable

HC-2000 RS232 serial (DCE)	Cable connects HC-2000 serial pin to PC serial pin	PC RS232 serial (DTE)	Stefan's cable colors
DB9 female		DB9 male	
1 – CTS (output)	8	1 – DCD Data Carrier Detect (input)	white
2 – DTR (input)	4	2 – RxD Receive Data (input)	purple
3 – RxD (output)	2	3 – TxD Transmit Data (output)	green
4 – TxD (input)	3	4 – DTR Data Terminal Ready (output)	orange
5 – Net		5- GND	brown
6 – GND	5	6 – DSR Data Set Ready (input)	grey
7 – GND		7 – RTS Request to Send (output)	blue
8 – GND		8 – CTS Clear to Send (input)	yellow
9 – N.C.		9 – RI Ring Indicator (input)	red

According to HC-2000 IF1 manuals, pin 1 CTS (output) of the HC-2000 (DCE device) should be connected to pin 8, CTS (input) of the DTE device (e.g., PC); initially, when doing so, the serial connection stopped working. However, after additional experimentation and development of the serial communication software it was determined that the behaviour was normal within parameters of the RTS/CTS flow control.

Also, also according to the HC-2000 IF1 manuals, pin 6 DSR of the DTE device is to be tied to +12V. Currently, this was determined to not be essential for the success of communication.

On the other hand, the Spectrum Microdrive Book in the RS232 section notes: "it follows that if two Spectrum are linked together by the RS232 system that **RXData**, **DTR**, **TxData** & **CTS** of one Spectrum will be connected to **TxData**, **CTS**, **RxData** & **DTR** respectively, of the second Spectrum". This configuration does not apply here since the communication is between a PC with a standard RS232 serial port and ZX spectrum compatible equipped with IF1 serial port.

Further, Spectrum Microdrive Book specifies that “for communication in both directions the user has to connect the **TXdata**, **RXdata**, **DTR**, **CTS** & **GND** of the Spectrum to the **data out**, **data in**, **RTS**, **CTS** & GND of the BBC microcomputer, respectively. The baud rate will be ‘9600’ by default on both machines.” This is indeed the case, the default baud rate is 9600 and any higher rate communication will require an additional command (FORMAT *”b” 19200 for example) to change the default value. In fact, after solving the hardware configuration problems, it was observed that the mismatch between baud rates was the most likely reason for communication failure. The other factor that led to communication failures was the buffering of serial data on the PC hardware which sometimes required the application to be restarted in order to flush or otherwise reset the serial port buffers.

On the HC-2000 it was observed that pin 1 CTS (output on DCE device) normally goes HIGH when waiting for data or when reading from the serial interface. This is the intended behaviour and further experimentation has determined that the CTS (output on DCE device) pin should be indeed connected to the CTS (input on DTE device). In addition, the RTS (output on DTE device) should also be connected to pin 2 DTR (input on DCE device) as illustrated above. This completes the configuration and enables the RTS/CTS flow control where one device will assert and read the appropriate signals and will not begin the actual transfer until the other device is ready to receive.

Finally, some considerations are necessary regarding the networking hardware of the IF1. According to the documentation, at least some of the hardware components are shared between the Network and Serial communication hardware. Currently no experimentation was done with the networking functions despite commands being very similar to those for serial communication. Therefore, HC-2000/IF1 network communication has yet to be explored and the relation to the serial communication work described here is currently unknown.

Testing serial transmission of text data

After establishing a physical serial connection between HC-2000 and the PC, a serial communication software must be run on the PC with the appropriate COM port open using a baud rate of 9600, 8 data bits, no parity and 1 stop bit. Then, the following Basic code should be run on the HC-2000.

```
10 FORMAT "t",9600
20 OPEN #3,"t"
30 LLIST
40 CLOSE #3
```

The result is that the code above will be transmitted and listed on the PC screen from where it can be manipulated further (copied, pasted, printed, etc).

Serial communication of text data is therefore easily possible from HC-2000 toward the PC which behaves just like a printer.

Testing serial transmission of binary data

The following short Basic program will send a binary data block with the HC-2000 screen content to the serial port where, in theory it can be received and loaded by another HC2000 or, alternatively, by a PC equipped with an adequate program that can store and interpret the binary data.

```
 1 FORMAT "b",9600
```

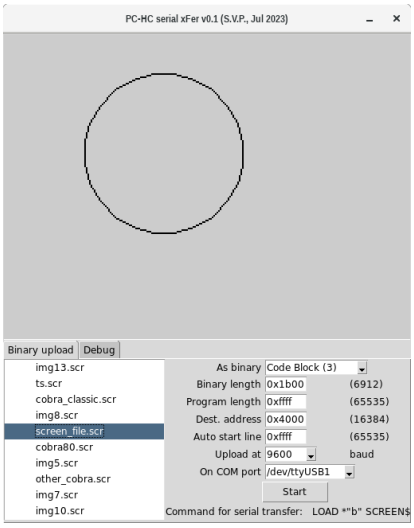
```
10 CIRCLE 100,100,50
15 SAVE *"b"SCREEN$
```

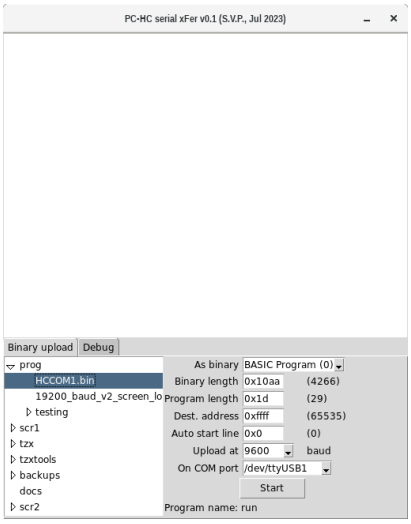
Using the RTS/CTS flow control wiring (see above) and appropriate programs on both PC and HC, binary data transfer from HC to PC and viceversa was possible without any apparent problems at 19200 baud. While theoretic baud rate are possible to in HC using the POKE 23747,INT(3500000/(26*BAUD_RATE))-2) command, higher rates are likely to result in transmission errors.

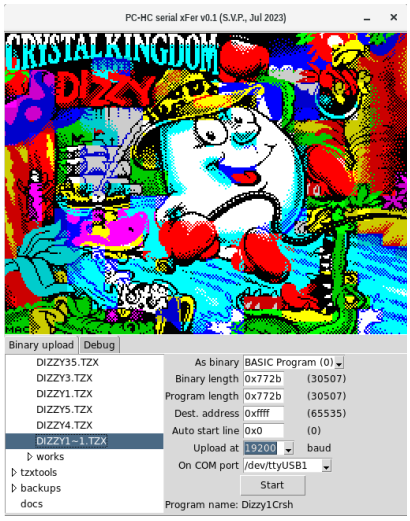
PC to HC (IF1) serial transmission of binary data

A software application was developed in Python to allow testing of serial transmission through the serial port in the RTS/CTS flow control configuration described earlier. Currently, using the application, it is possible to transfer screen data at the maximum rate of 19200 baud. Code blocks and Basic program data in form of simple binary files TXZ files can also be transferred, so long as they do not contain multiple blocks or rely on custom loaders.

In order to increase usability, in addition to reading binary file sizes, code was included to read binary file headers (when available) or the information in the TZX header blocks in order to prepopulate the user interface with the appropriate values for program length, destination address, autostart line, etc .







The first screenshot shows the UI after selecting a screen binary file whose upload parameters are immediately determined and the corresponding bitmap image displayed. The screen data can be uploaded using LOAD *”b” SCREEN$ command.

The next screenshot shows the HCCOM1.bin binary image of the HC Commander program that is loadable through the serial port using the Basic command LOAD *”b”. A separate header file *.binh (not visible in the UI, but nonetheless present in the file system) provides the additional upload parameters (program length, auto start line number) of the Basic program that is to be loaded.

The third screenshot shows the TZX file of a game that can be loaded with the same extended Basic command. Currently, this is possible only for TZX files that are already modified to contain at most

two standard speed data blocks (header + code) in order to facilitate loading from any media. The game art in this third screenshot is only for illustrative purposes, it does not come from the TZX file itself.

Conclusions

Why the effort to investigate serial data transfer? Not only this allows connecting HC-2000 to other serial devices and extending its functionality (e.g., as a TELNET terminal) it also opens possibilities to easily and automatically upload data and programs, unlocking more sophisticated interaction between machines. This will facilitate development of software by reducing the amount of effort necessary to quickly deploy and test software application prototypes.

Currently, serial transfers are limited to single-duplex whereby communication occurs in only one direction at a time using explicit LOAD and SAVE commands. Additional software and packet-oriented and socket-oriented protocols could in theory take more advantage of the existing hardware in order to enable full-duplex transfers, error checking, re-transmit, similar to how TCP/IP software stacks work.

Finally, it has to be acknowledged that there is some variability between serial port hardware. It was noted that while PC transfer application that changed baud rate dynamically (while the port was open) worked well with a serial adapter (067b:2303 Prolific Technology, Inc. PL2303 Serial Port). The same capability was not present or was difficult to achieve with a built-in RS232 on a PC motherboard. In that case, transfers were limited to one baud rate value unless the port was closed and reopened.

References

1. HC-2000 manual Tehnic si de Utilizare; ICE Felix Computer S.A.
2. Interfata 1 HC; Manual de Utilizare; ICE Felix Computer S.A.
3. SPECTRUM MICRO DRIVE BOOK / with details of the ZX Interface 1; the Microdrive, the Local Area Network and the RS232 Link by Dr. lan Logan; Melbourne House Software Inc., 1983
4. Game art - in form of screen files to test image data transfer; https://zxart.ee/eng/graphics/games/
5. HC Commander, version 1.0, George Chirtoaca https://github.com/0sAND1s/HCCmd

```
10 FORMAT \"t\";9600
20 OPEN #3,\"t\"
30 LLIST
35 CAT #3,1
37 INPUT #3;a$
38 PRINT a$
40 CLOSE #3
```