



## Forum

Discussions sur les vieux ordinateurs, pockets &amp; consoles

Rechercher...

[Raccourcis](#) [FAQ](#)[Inscription](#) [Connexion](#)[Site](#) < [Accueil](#) < [MySilicium](#) < [Silicium in English](#)

## SHARP:USB-cable HW-handshake

Modérateur : Politburo

Répondre



Rechercher...



14 messages • Page 1 sur 1

## SHARP:USB-cable HW-handshake

par **spellbound** » 09 oct. 2017 23:40

This post is about how to obtain a reliable, full speed connection between a SHARP pocket computer that is equipped with an RS-232 interface and a PC/MAC. It's about utilizing bidirectional (RTS/CTS) hardware handshake so you do not need the XON/XOFF-protocol, line-delays or lowering baud-rates at all.

SHARP pockets to which this applies are (at least):

- PC-1600
- PC-E500 series
- PC-13xx series
- PC-1450 / 1475
- PC-E220
- PC-G8xx series
- ...

or in general every SHARP pocket with a 15pin or 11pin RS-232 interface.

There are already many solutions out there but none with a true one-fits-all approach.

Furthermore the wirings I've seen for the E500 and 1350|60 all require the activation of the XON/XOFF-protocol for LOADING.

Especially fixed wirings/cables that offer bidirectional hardware handshake for the PC-1600 in principle cannot do the same for the E500 or 13xx series and vice versa, because there are subtle but relevant

**spellbound**

Fonctionne à 75 bauds



Messages : 74

Inscription : 06 mai 2015 12:06

differences.

In fact, I found that there are three different types of built-in RS-232 interfaces in the family of SHARP pockets:

- PC-1600
- 15pin Standard
- 11pin Standard

### USB-Adaptor

Lets start with the USB-side of an adaptor/cable, which can be the same for all three types:

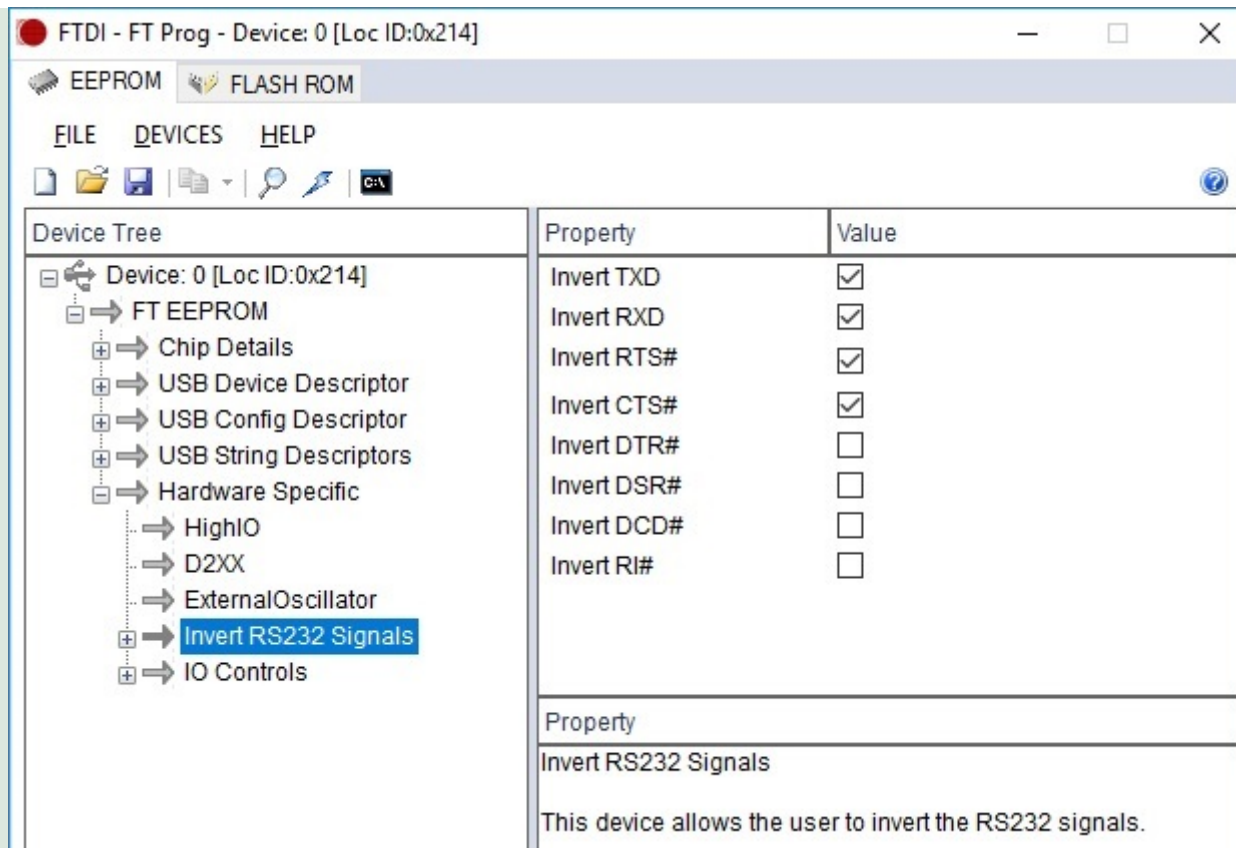
I recommend an FTDI USB-adaptor/cable with the FT232R-chip, 5V, open ended.



### Setup FTDI

Download the tool FT\_PROG from the website of the manufacturer: [www.ftdichip.com](http://www.ftdichip.com).

With this tool you must logically invert the signals RXD, TXD, RTS and CTS of the FTDI-chip, because the FTDI-chip exposes UART-TTL signal levels, but the RS-232 interface of the SHARP pockets operates on inverted UART-logic. This is a one-off process and the settings are persisted within the integrated EEPROM of the FTDI-chip.



So you now have a cable that exposes the following signals with inverted UART logic, 5V HIGH:  
GND, VCC, RXD, TXD, CTS, RTS

Now lets care about the specifics of the three interface types mentioned above and appropriate wirings and settings for bidirectional hardware handshake.

### PC-1600 Interface

The port pinout of that machine is as follows:

SHARP PC-1600 15-Pin RS-232C Port				
Pin#	Signal	Signal Full Name	Voltage Level	Direction
1	FG	Frame Ground (Shield)	none	none
2	TXD	Transmitted Data	LOW: -8.5V, HIGH: +5.6V	out
3	RXD	Received Data	LOW: <=0V, HIGH: >= +3V	in
4	RTS	Request To Send	LOW: -8.5V, HIGH: +5.6V	out
5	CTS	Clear To Send	LOW: <=0V, HIGH: >= +3V	in
6	DSR	Data Set Ready	LOW: <=0V, HIGH: >= +3V	in
7	GND	Ground	0V	none
8	CD	Carrier Detect	LOW: <=0V, HIGH: >= +3V	in
9	CI	Call Indicator	LOW: <=0V, HIGH: >= +3V	in
10	VCC	Voltage Supply	+4V to +4.7V	none
11	-	-	-	out
12	-	-	-	in
13	-	-	-	none
14	DTR	Data Terminal Ready	LOW: -8.5V, HIGH: +5.6V	out
15	-	-	-	out

This port can handle standard RS-232 input signal levels

In addition to the RS-232C standard, the valid input signal range for LOW is extended (from -3V) up to 0V

=> All inputs accept inverted UART TTL signal levels



RS-232C Port

SIO-Port

As you can see, the TXD and RTS outputs of the PC-1600 are -8.5V LOW and +5.6V HIGH.

The PC-1600 is the only SHARP-pocket, that provides true RS-232 signal levels,

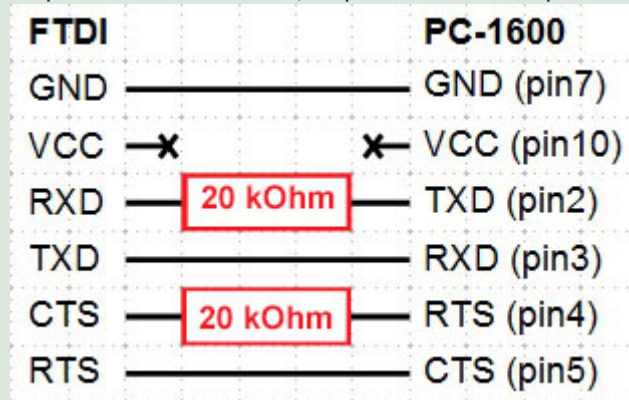
which is the reason for the CE-1601L - CE-1605L being just cables and not level shifters like the CE-130T - CE-133T and others.

But the FTDI-chip seems to have internal clipping of negative voltages, so we can use a straight forward null-modem wiring, DTR/DSR/CD can be ignored.

<EDIT>

There are some FTDI-remakes out there, that seem to require additional resistors (10-100 kOhm) in series

for the outgoing signals of the PC-1600, which are RTS and TXD. The original FTDI-USB cable does not require those resistors, depicted in red (replace them by a direct wiring in the following diagram).



Within the rest of this article I'm ignoring the additional resistors for those FTDI-remakes.

Setup PC-1600:

CODE : TOUT SÉLECTIONNER

```
SETCOM "COM1:",9600,8,N,1,N,N
SNDSTAT "COM1:",59
RCVSTAT "COM1:",61
OUTSTAT "COM1:"
```

The SNDSTAT, RCVSTAT and OUTSTAT statements are mandatory! – they activate the RTS/CTS hardware handshake for both directions.

Setup Terminal Program (CoolTerm, hterm, ...)

- baud = 9600
- data bits = 8
- parity = none
- stop bits = 1
- XON/XOFF = off
- RTS/CTS (RS/CS) = on
- line delay = off
- character delay = off
- RTS (initial) output (button/toggle) = ON/HIGH (deactivate to pause transmission from the pocket)

Thats it for the PC-1600 😊

Unfortunately this wiring does NOT provide bidirectional hardware-handshake for the 15pin SHARP standard interface!

### **15pin Standard Interface**

SHARP pockets with this type of interface are the following:

PC-E500 series, PC-1350, PC-1360, PC-1475 and all others with 15pin RS-232 interface except the PC-1600.

There are two relevant differences between the 15pin PC-1600 interface and the 15pin standard interface

- Voltage levels are 0V LOW and 5V HIGH
- The meaning and behavior of the RTS signal

In the mid to late 1980's there was a transition in the industry regarding the interpretation of the RTS signal of the RS-232 standard towards the new RTR meaning (which was not RS-232 standard conformant).

The original, historical meaning was: RTS = "Request To Send", i.e. the computer wants to send data out and requests permission from the connected device to do so via RTS. The device then answers by CTS. The problem is, that this protocol is asymmetrical (RTS and CTS are not independent) and the computer has no means to signal the connected device to pause a data transmission going from the device to the computer, in case the latter is busy.

This soon became a big problem in computer-to-computer communication. What was/is really needed is this: By RTS the computer allows the connected device to send data to the computer. In other words this new interpretation has the meaning of "Ready To Receive" (RTR). RTR and CTS have a symmetrical meaning (i.e. supporting both directions) and are independent from each other. In fact today the meaning of RTS has "quietly" been shifted to RTR, although the label RTS has been kept.

This is also true for the RTS-signal of the PC-1600!

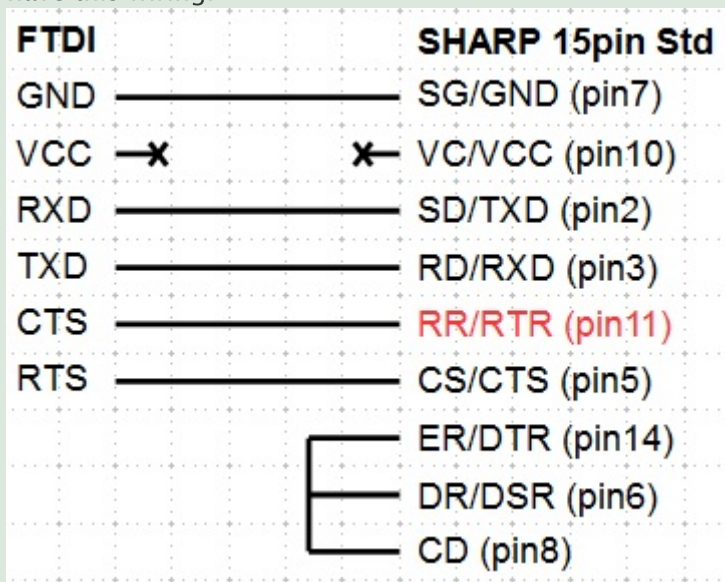
From the pinout below we can read that in the mid 80ies SHARP decided to deal with this problem by providing the original standard conformant RTS(RS) signal *as well as* an additional RTR(RR) signal for it's standard 15pin interface:



SHARP Standard 15pin RS-232 Port			
Pin#	SHARP Signal Name	SHARP Full Name	Direction
1	FG	Frame Ground	none
2	SD	Send Data	out
3	RD	Receive Data	in
4	RS	Request to Send	out
5	CS	Clear to Send	in
6	DR	Data Set Ready	in
7	SG	Signal Ground	none
8	CD	Carrier Detect	in
9	CI	Call Indicator	in
10	VC	Voltage Supply	none
11	RR	Ready to Receive	out
12	PAK (CE-140P)	Periph. Acknowledge	in
13	VC	Voltage Supply	none
14	ER	Equipment Ready	out
15	PRQ (CE-140P)	Periph. Request	out

So for a wiring that supports bidirectional hardware handshake, we need the RTR(RR) signal and NOT the RTS(RS) signal from this type of interface!

Additionally at least the PC-E500 series requires a DTR/DSR/CD handshake in order to activate the interface. So we need a respective loopback for that. That means for the SHARP 15pin standard interface we basically have this wiring:



Setup PC-1350/60 for bidirectional hardware handshake:

CODE : TOUT SÉLECTIONNER
OPEN "1200,N,8,1,A,C,&1A"
CLOSE

Setup PC-E500(S) for bidirectional hardware handshake:

CODE : TOUT SÉLECTIONNER
OPEN "9600,N,8,1,A,C,&H1A,N,N"
CLOSE

These settings are persistent (i.e. stored until explicitly changed or the next hard reset is performed).  
If you OPEN the port without any parameter string, the stored settings will be used:

CODE : TOUT SÉLECTIONNER
OPEN
LOAD / SAVE
CLOSE

The terminal program config is as above (except the max baud rate of 1200 for e.g the 13xx)

### **11pin Standard Interface**

This type of RS-232 interface is the latest in the line of SHARP pocket computers, so it is not astounding that its RTS-signal already has RTR semantics.

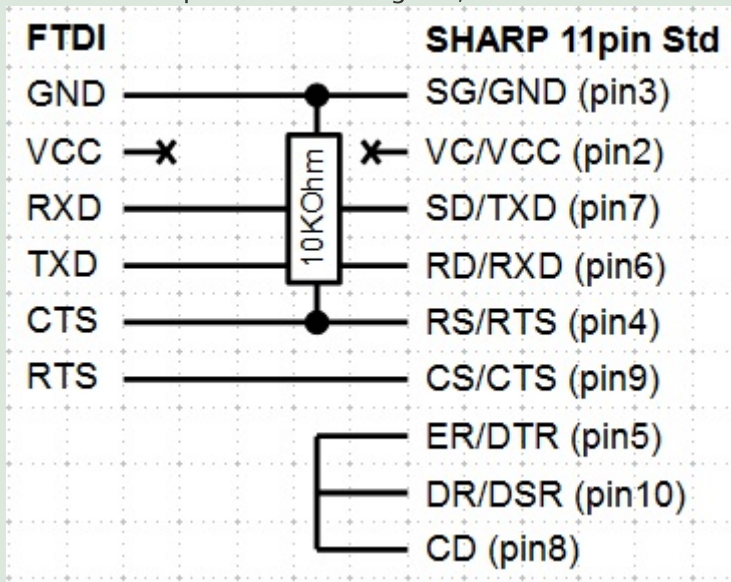
Pockets with this type of interface are: PC-E220, PC-G850V(S) and all other with an 11pin RS-232 interface.

Typically the 11pin interface supports different operational modes – here is the pinout for RS-232 mode:



SHARP Standard 11pin Port in RS-232 Mode				
Pin#	SHARP Signal Name	Alternative Name	Signal Full Name	Direction
1	-		-	-
2	VCC	VC	Voltage Supply	none
3	GND	SG	Ground	none
4	RTS	RS	Request To Send	out
5	DTR	ER	Data Terminal Ready	out
6	RXD	RD	Received Data	in
7	TXD	SD	Transmitted Data	out
8	CD		Carrier Detect	in
9	CTS	CS	Clear To Send	in
10	DSR	DR	Data Set Ready	in
11	CI		Call Indicator	in
Pin1 = leftmost, pin11 = rightmost, when viewed at the pocket computer interface				

The PC-G850V(S) sets DTR to HIGH when the interface is activated, but does not care about DSR and CD. On the other hand, a DTR/DSR/CD loopback is not harmful, so lets include it in the wiring, just to be safe. Additionally, for the PC-G850V I found it necessary to incorporate a 10KOhm pulldown resistor to the RTS signal. Without that, the host computer has no defined LOW level and does not pause data transmission, when the G850V requests it – leading to I/O errors.

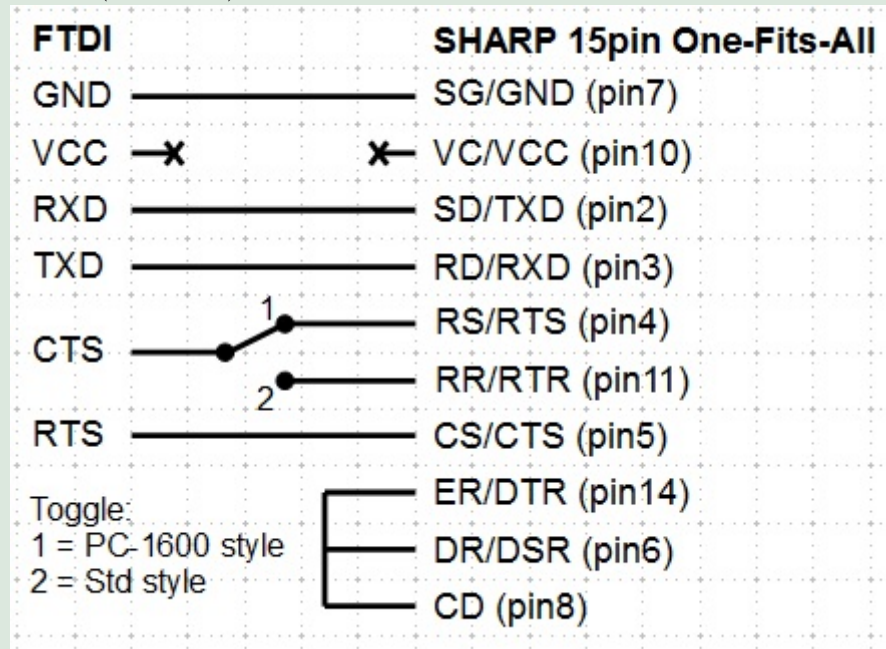


PC-E220 and PC-G850 setup:  
Menu TEXT->Sio->Format

baud rate = 9600  
 data bit = 8  
 stop bit = 1  
 flow = RS/CS

### One-Fits-All 15pin

If you want to build an adapter/cable that supports the PC-1600 hardware handshake as well as the 15pin standard, you obviously have to merge the two wirings from above. And that leads to the necessity of a toggle switch (or similar):



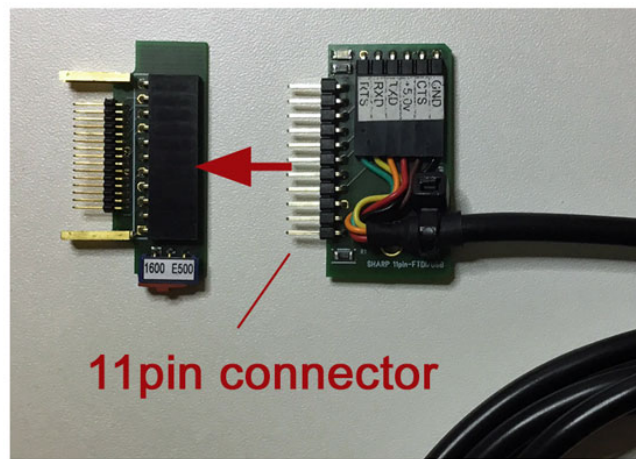
### 11pin to 15pin Adaptor

If you want a solution that supports all three types of interfaces, you could build a modular adaptor with a removable 11pin-to-15pin adapter part that has a 1:1 signal mapping and incorporates the toggle switch. The DTR/DSR/CD loopback and the RTS-pulldown resistor then remains on the part that is attached to the cable.

Here is a picture of how this could look like:



15pin connector



11pin connector

I hope this was inspiring and informative.  
Enjoy  
Tom

Dernière édition par spellbound le 17 oct. 2021 20:03, édité 10 fois.

## Re: SHARP:USB-cable HW-handshake

par **pir2** » 15 oct. 2017 10:12

THanks for this detailed explanation, i will see if this is applicable to the 2 USB-cables i've bought some years ago .... and never really tested 😊



**pir2**

Fonctionne à 9600 bauds



Messages : 4623

Inscription : 31 oct. 2006 16:08

Localisation : 67310 Westhoffen

Contact :

## Re: SHARP:USB-cable HW-handshake

par **spellbound** » 15 oct. 2017 13:58

Hopefully yes – good luck 😊

Dernière édition par spellbound le 31 déc. 2018 20:45, édité 1 fois.



**spellbound**

Fonctionne à 75 bauds



Messages : 74

Inscription : 06 mai 2015 12:06

## Re: SHARP:USB-cable HW-handshake

par **philoupat** » 30 sept. 2018 20:58

Bonjour

j'ai réalisé une connexion usb avec mon pc1350

j'utilise le programme suivant

5 "A"

10 OPEN "1200,N,8,1,A,C,&1A"

20 LPRINT "BONJOUR"

j'utilise Hterm

La commande Run fonctionne bien

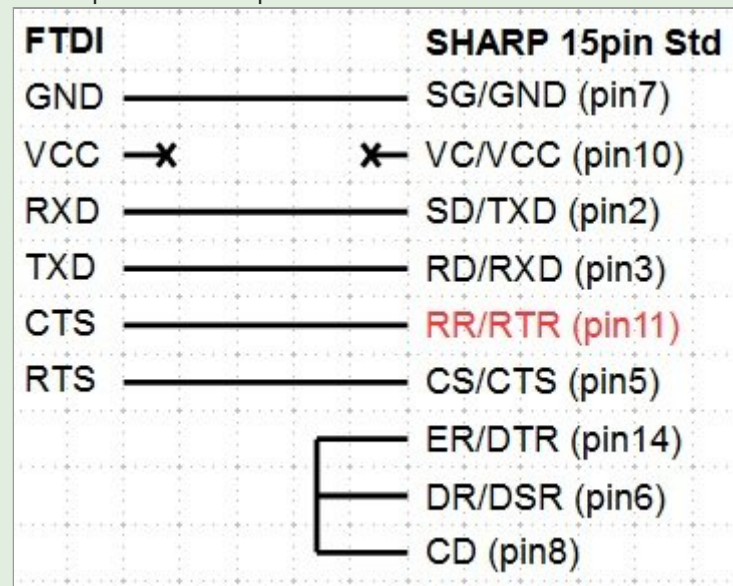
Les commandes OPEN et LLIST fonctionnent bien (voir ci-dessus)

ma question est la suivante : comment envoyer un fichier programme dans la machine

la commande LOAD BRK ne donne rien

La commande SAVE BRK ne donne rien

merci pour votre réponse



usb ftdi

usb.jpg (33.76 Kio) Consulté 29555 fois

La pièce jointe **Capture.PNG** n'est plus disponible



**philoupat**

Fonctionne à 75 bauds



Messages : 12

Inscription : 11 mars 2018 22:32

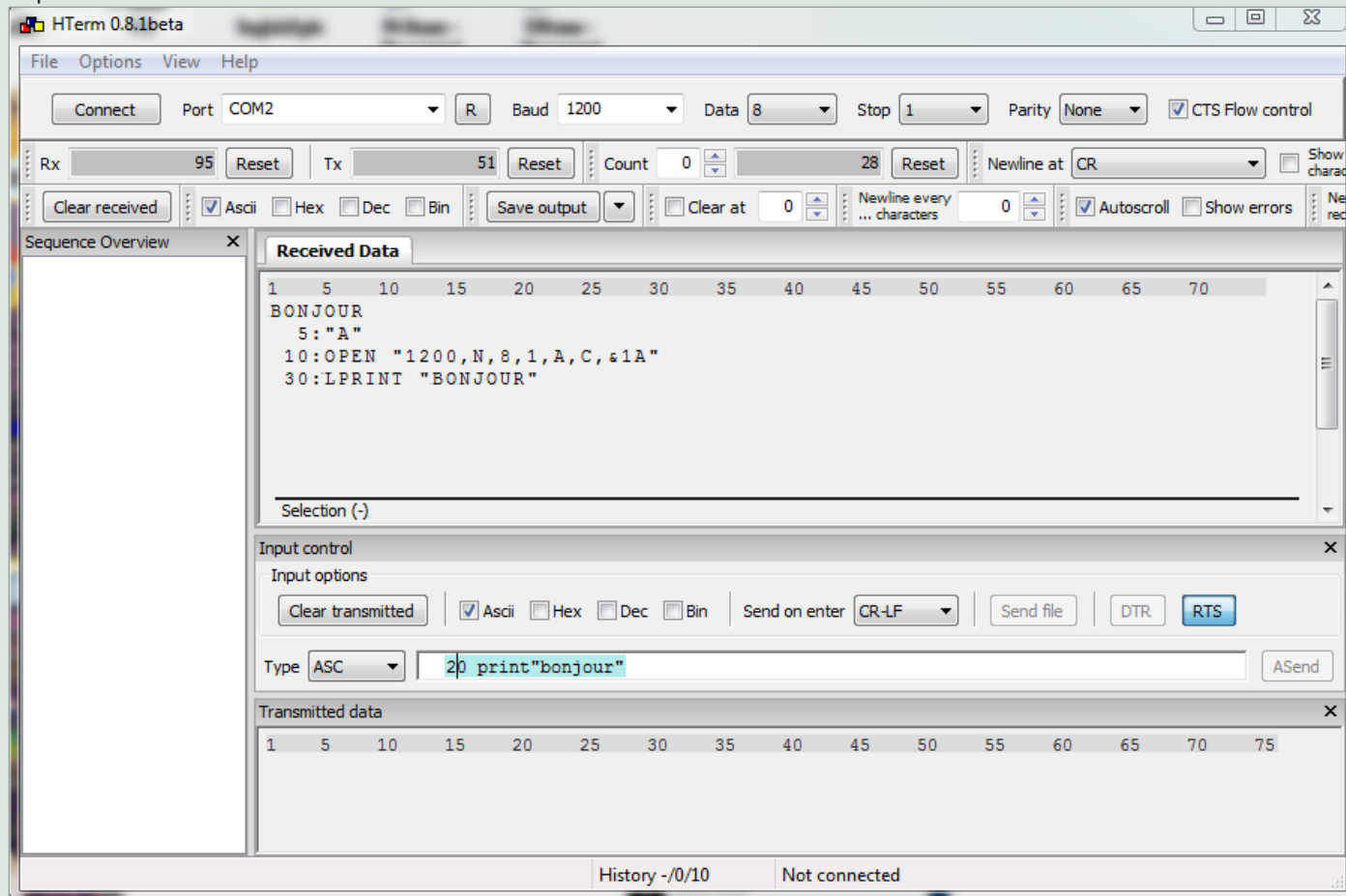
## Re: SHARP:USB-cable HW-handshake



**philoupat**

par **philoupat** » 30 sept. 2018 20:59

## capture de hterm



Capture.PNG (87.9 Kio) Consulté 29554 fois

Fonctionne à 75 bauds



Messages : 12

Inscription : 11 mars 2018 22:32

## Re: SHARP:USB-cable HW-handshake

par **spellbound** » 25 oct. 2018 22:22

Upload a file from PC:

In hterm:

1. press the „Connect“ button
2. press the „Send file“ button (located at the Input Control section)
3. in the dialog select the file you want to upload from your hard disk (do not press „Send“ by now)



**spellbound**

Fonctionne à 75 bauds



Messages : 74

Inscription : 06 mai 2015 12:06



At the 1350:

4. OPEN

5. LOAD (this command will block, do not press BRK – it waits for the file to be send)

In hterm again:

6. press „Send“ in the file dialog and wait until the transmission has completed

Dernière édition par spellbound le 27 oct. 2018 18:26, édité 1 fois.

## Re: SHARP:USB-cable HW-handshake

par **spellbound** » 27 oct. 2018 15:21



A rather tricky part when performing file uploads of programs found in the internet, is the correct EOL (End Of Line) and EOF (End Of File) handling.

Since many people struggle at this point and asked for help, I give some general advice here.

The essence is that the EOL and EOF setting of your (SHARP pocket) computer must match the actual format of the file to upload.

If the EOL format does not match you will get a transmission error.

If the EOF token does not match or does not exist at the end of your file, the transmission will not terminate (at least when using full hardware handshake, like suggested in this thread).

Note that this compatibility issue does not arise, if you just LOAD a program that you have previously SAVED with the same settings.

It may arise, when you try to transfer program source-code that has been SAVED with a different computer or has been downloaded from the internet.

It will arise, when you try to transfer program source-code that has been SAVED with different EOL/EOF-settings.

To check or modify the actual file format, I recommend downloading the free source-code editor Notepad++.  
The following diagram summarizes all necessary steps:

**spellbound**

Fonctionne à 75 bauds



Messages : 74

Inscription : 06 mai 2015 12:06

## EOL/EOF Settings for File Upload to SHARP Pocket Computer

### Notepad++ Editor

-> Main-Menu -> View -> Show Symbol -> Show End of Line

### SHARP EOL Notation

**C** (= CR only)

```
14 390 GOTO 450CR
15 400 X=X+2: GOTO 450CR
16 410 X=X-2:Y=Y-2: GOTO 450CR
17 420 Y=Y-2: GOTO 450CR
18 430 X=X+2:Y=Y-2: GOTO 450CR
19 440 IF Y<1 OR Y>30 GOTO 490CR
20 450 PSET (X,Y)CR
```

Convert file to this format with Notepad++:

-> Main-Menu -> Edit -> EOL Conversion

-> Old MAC Format

**F** (= LF only)

```
14 390 GOTO 450LF
15 400 X=X+2: GOTO 450LF
16 410 X=X-2:Y=Y-2: GOTO 450LF
17 420 Y=Y-2: GOTO 450LF
18 430 X=X+2:Y=Y-2: GOTO 450LF
19 440 IF Y<1 OR Y>30 GOTO 490LF
20 450 PSET (X,Y)LF
```

-> UNIX/OSX Format

**L** (= CR+LF)

```
14 390 GOTO 450CRLF
15 400 X=X+2: GOTO 450CRLF
16 410 X=X-2:Y=Y-2: GOTO 450CRLF
17 420 Y=Y-2: GOTO 450CRLF
18 430 X=X+2:Y=Y-2: GOTO 450CRLF
19 440 IF Y<1 OR Y>30 GOTO 490CRLF
20 450 PSET (X,Y)CRLF
```

-> Windows Format

### SHARP RS-232 file format definition commands:

PC-1350/60, PC-1450/75:

OPEN "1200,N,8,1,A,C,&1A"

☐ EOL (End Of Line) setting

☐ EOF (End of File) setting

PC-1600:

Fixed EOL setting "L" (i.e. Windows format) for file transfer!

Fixed EOF setting "&1A"!

The PCONSOLE command applies to LPRINT only, not to file transfer!

PC-E500/S:

OPEN "9600,N,8,1,A,C,&H1A,N,N"

Notepad++ displays the hexadecimal value "1A" as the special token "SUB"

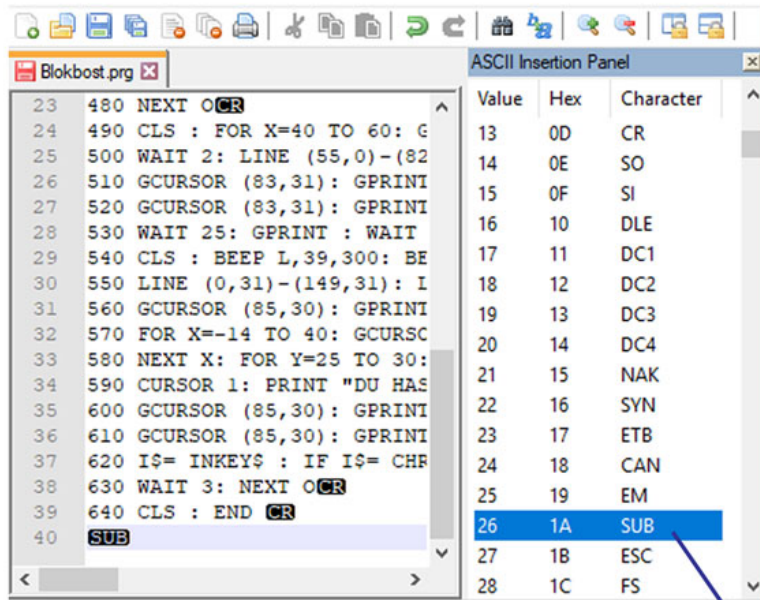
```
38 630 WAIT 3: NEXT OLF
39 640 CLS : END LF
40 SUB
```

typical EOF marker (SUB = &1A) for SHARP pocket computer

Activate the Notepad++ "ASCII Insertion Panel" to append a missing EOF token to the end of a file:



Main-Menu -&gt; Edit -&gt; Character Panel



Click here to insert a SUB (=hex1A) token

As you can see, with a PC-1350 for example, you can adjust the settings at your pocket, to match the actual file format. With a PC-1600 you can't, which means you have to convert the file format, if it does not match (check and change respectively with Notepad++).

Dernière édition par spellbound le 17 janv. 2021 14:41, édité 1 fois.

## Re: SHARP:USB-cable HW-handshake

par **dogrin17** » 11 août 2020 03:26

Would anyone know why my G850VS can receive data just fine, but sits on "SENDING" when trying to send? I get no data in the terminal. I've followed all of the steps and I'm a bit lost.

Can anyone help? Thank you.



**dogrin17**

Fonctionne à 75 bauds



Messages : 3

Inscription : 11 août 2020 01:22

## Re: SHARP:USB-cable HW-handshake

par **spellbound** » 14 août 2020 23:55



**spellbound**

Fonctionne à 75 bauds



Messages : 74

“ dogrin17 a écrit : ↑

11 août 2020 03:26

Would anyone know why my G850VS can receive data just fine, but sits on "SENDING" when trying to send? I get no data in the terminal.

That means the G850VS is blocked by the RTS-signal from your computer. The most likely reason for that is that you forgot to activate the signal from within your terminal software (see screenshot of hterm above – the RTS toggle button in the input control section has to be switched ,on‘)

## Re: SHARP:USB-cable HW-handshake

par dogrin17 » 15 août 2020 00:10



“ spellbound a écrit : ↑

14 août 2020 23:55

That means the G850VS is blocked by the RTS-signal from your computer. The most likely reason for that is that you forgot to activate the signal from within your terminal software (see screenshot of hterm above – the RTS toggle button in the input control section has to be switched ,on‘)

Thank you, it's working now. I really appreciate the detailed guide.

**dogrin17**

Fonctionne à 75 bauds



Messages : 3

Inscription : 11 août 2020 01:22

## Re: SHARP:USB-cable HW-handshake

par engelen » 14 févr. 2023 23:58



After constructing an FTDI adapter for a PC-G850VS, I found that the FTDI driver 1.5.0 for Mac M1 machines fails to respect the FTDI CTS input. Without respecting the handshake, the PC keeps sending to the PC-G850VS and data will get lost. The FTDI drivers for Windows and Mac Intel machines work fine, but the driver for Mac M1 fails. Perhaps a bug.

Also, what is your recommendation with respect to 3.3V versus 5V FTDI adapters? I used a FTDI 5V adapter (Adafruit FTDI Friend) but it didn't work to send from the PC-G850VS to the PC or Mac (with inverted TXD, RXD, CTS, RTS and with CTS/RTS handshake.) Strangely, when I manually disable DTR then it works, which should have nothing to do with CTS/RTS handshake. On the other hand, there is no problem at all with the FTDI 3.3V adapter, but the 5V gives this strange behavior to block sends from the PC-G850 unless DTR is cleared manually.

Here is a longer post on the HP forum: <https://www.hpmuseum.org/forum/thread-1 ... #pid169250>

– Robert

**engelen**

Fonctionne à 75 bauds



Messages : 20

Inscription : 01 sept. 2021 21:14

Localisation : New York

Contact :

## Re: SHARP:USB-cable HW-handshake

par **spellbound** » 17 févr. 2023 23:14

For the G850V/S the RTS-pulldown resistor mentioned above is essential (10–100kOhm). Without it the G850 can't reliably pause transmissions from the host, just as you experienced.



**spellbound**

Fonctionne à 75 bauds



Messages : 74

Inscription : 06 mai 2015 12:06

## Re: SHARP:USB-cable HW-handshake

par **engelen** » 03 mars 2023 22:32



For the G850V/S the RTS-pulldown resistor mentioned above is essential (10–100kOhm). Without it the G850 can't reliably pause transmissions from the host, just as you experienced.



**engelen**

Fonctionne à 75 bauds



Messages : 20

Inscription : 01 sept. 2021 21:14

Localisation : New York

Contact :

Yes, I already did that and have the resistor as shown in the pictures in the HP Forum thread. But after several hours tinkering I found that the FTDI driver for MacOS M1 machines has a bug. There is no other explanation. MacOS Intel works fine. Windows machines too. I also tried 3.3V and 5V FTDI cables and different FTDI feathers, but makes no difference (although 3.3V works best, since 5V may have an issue as described earlier). It's the damn driver. The latest FTDI driver for MacOS M1 machines fails. It doesn't respect RTS. Others may not have had this issue if they are using an older FTDI driver or a non-Mac M1.

– Robert

## Re: SHARP:USB-cable HW-handshake

par **spellbound** » 07 mars 2023 01:10



**engelen** a écrit : ↑

03 mars 2023 22:32

I found that the FTDI driver for MacOS M1 machines has a bug. There is no other explanation.  
– Robert

**spellbound**

Fonctionne à 75 bauds



Messages : 74

Inscription : 06 mai 2015 12:06

ok, I see

Répondre



14 messages • Page 1 sur 1

< Revenir vers « Silicium in English »

Aller

[🏠 Site](#) < **Accueil**

[🗑 Supprimer tous les cookies du forum](#) Le fuseau horaire est réglé sur UTC+02:00

Développé par phpBB® Forum Software © phpBB Limited

[PRIVACY\\_LINK](#) | [TERMS\\_LINK](#)