SHARP:USB-cable HW-handshake

Modérateur : Politburo

SHARP:USB-cable HW-handshake

par **spellbound** » lun. oct. 09, 2017 23:40 pm

spellbound
Fonctionne à 75 bauds
Messages : 67
Inscription : mer. mai 06, 2015 12:06 pm
Contact :

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-1475
- PC-E220
- PC-G8xx series
- ...

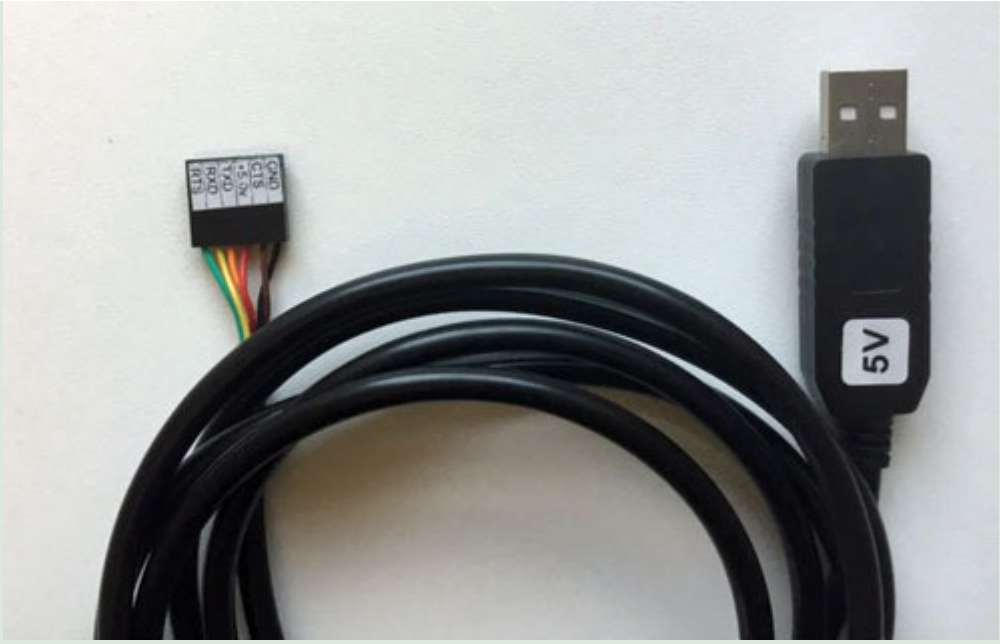
or in general every SHARP pocket with a 15pin or 11pin RS-232 interface.

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 subtile but relevant 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

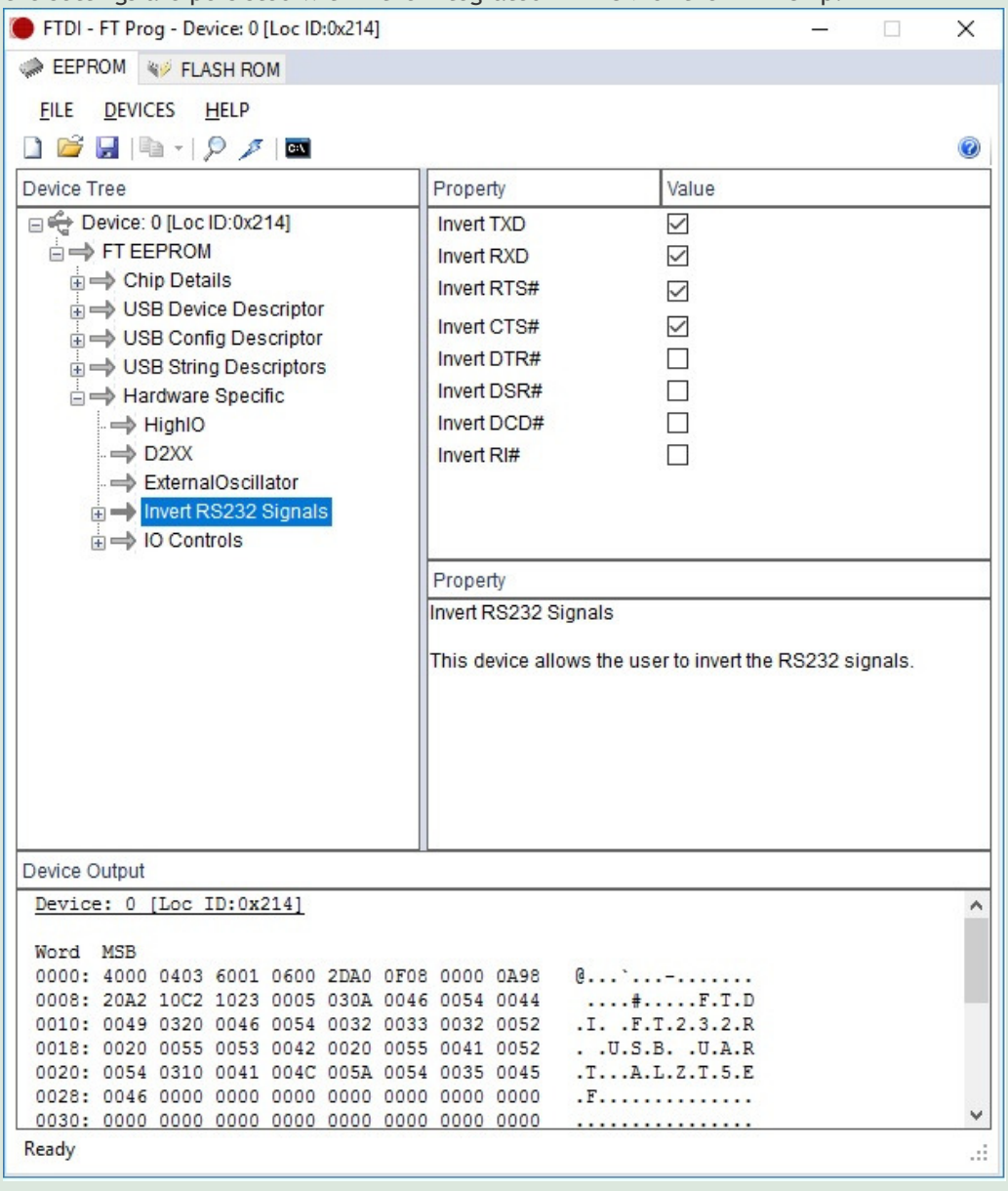
USB-Adaptor
Lets start with the USB-side of an adaptor/cable, which can be the same for all three types:
I recommend an FTDI USB-adaptor/cable with the FT232R-chip, 5V, open ended.



e.g. here: FTDI-FT232R-cable

Setup FTDI

Download the tool FT_PROG from the website of the manufacturer: 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

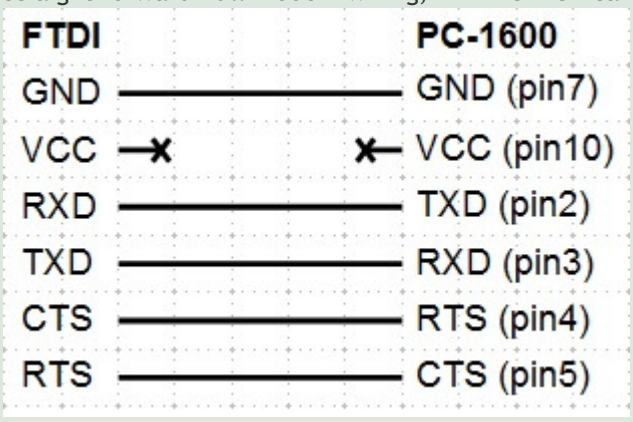
Now lets care about the specifcs of the three interface types mentioned above and appropriate wirings and settings for full hardware handshake.

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				

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 beeing 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.



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).

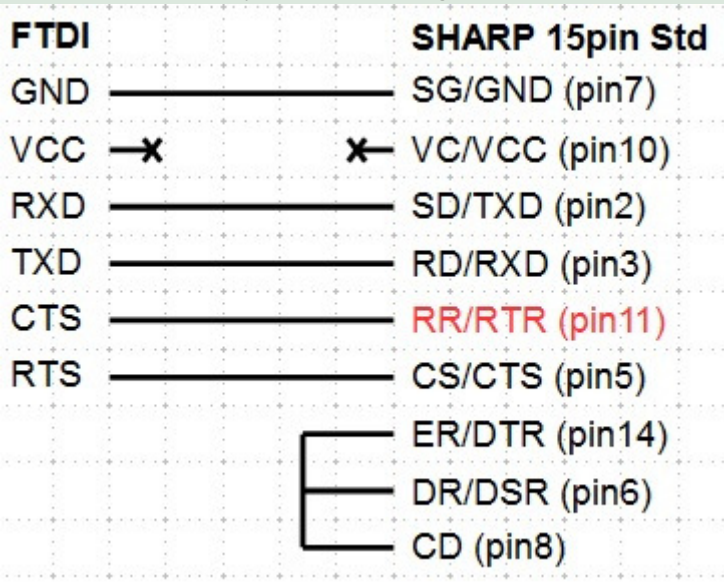
The original, historical meaning was: RTS = "Request To Send", i.e. the computer wants to send data out and requests permission form the connected device to do so via RTS. The device then answers by CTS. The problem is, that this protocol is asymmetrical (RTS and CTS are not independent) and the computer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

CODE : TOUT SÉLECTIONNER

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

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

CODE : TOUT SÉLECTIONNER

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

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

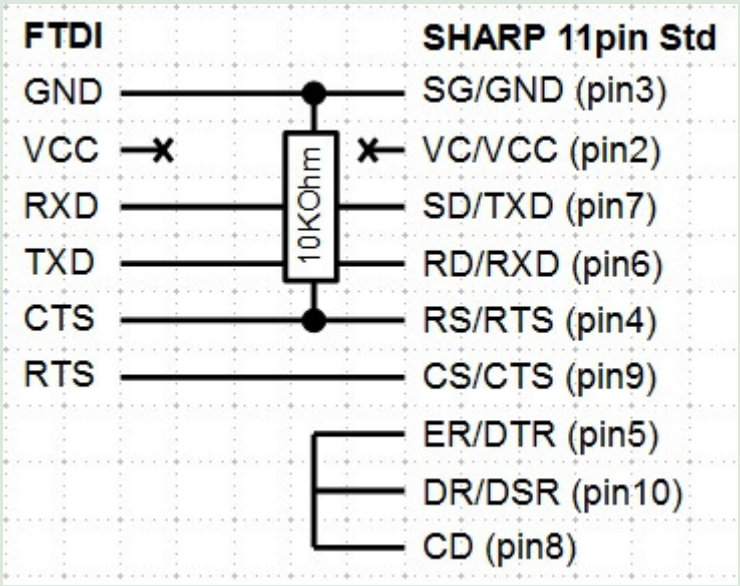
11pin Standard Interface

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:

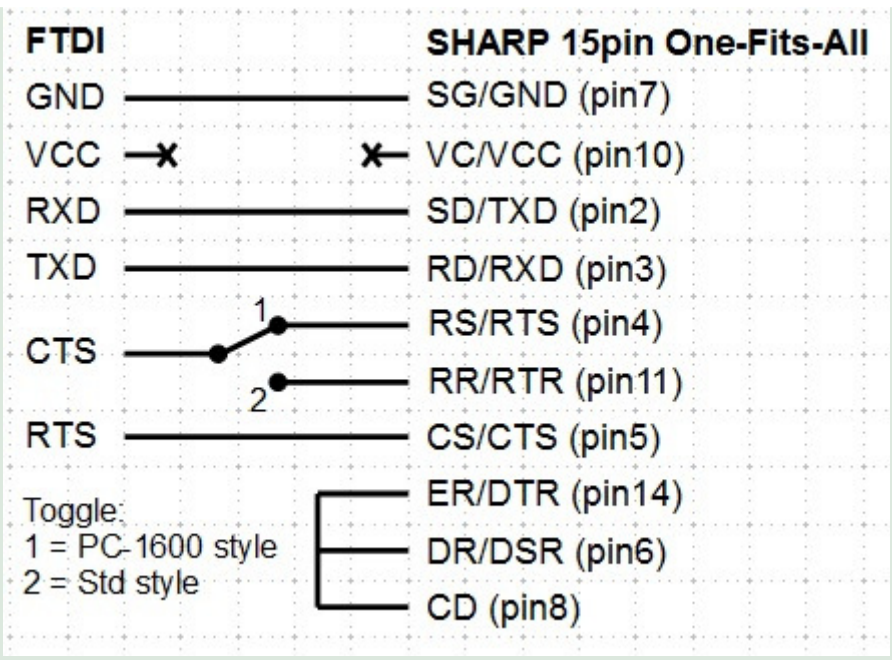
SHARP Standard 11pin Port in RS-232 Mode				
Pin#	SHARP Signal Name	Alternative Name	Signal Full Name	Direction
1	-		-	-
2	VCC	VC	Voltage Supply	none
3	GND	SG	Ground	none
4	RTS	RS	Request To Send	out
5	DTR	ER	Data Terminal Ready	out
6	RXD	RD	Received Data	in
7	TXD	SD	Transmitted Data	out
8	CD		Carrier 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

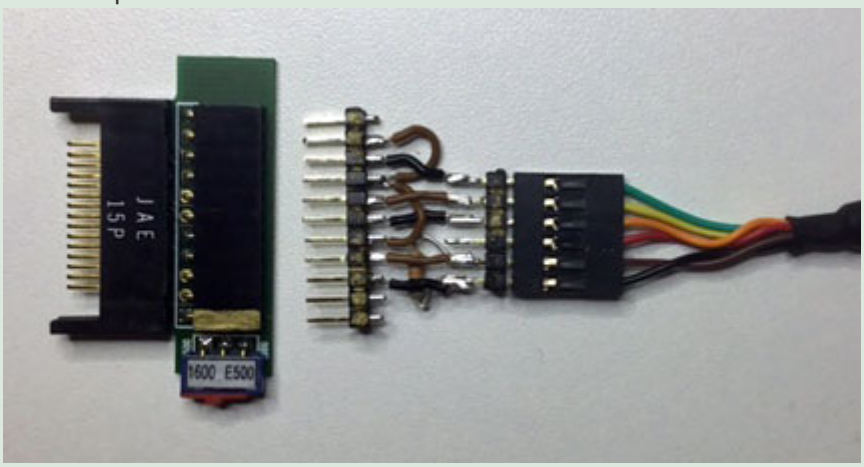
One-Fits-All 15pin
If you want to build an adapter/cable that supports the PC-1600 hardware handshake as well as the 15pin standard, you obviously have to merge the two wirings from above. And that leads to the necessity of a toggle switch (or similar):



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:



finally you can cover the open wirings by a shell or use a PCB.

I hope this was inspiring and informative.
Enjoy
Tom

Répondre

1 message • Page **1** sur **1**

‹ Revenir vers « Silicium in english »

Aller

QUI EST EN LIGNE ?

Utilisateurs parcourant ce forum : pir2 et 0 invité

Site ‹ **Accueil** L'équipe Membres Supprimer tous les cookies du forum Le fuseau horaire est réglé sur UTC+02:00

Nous contacter