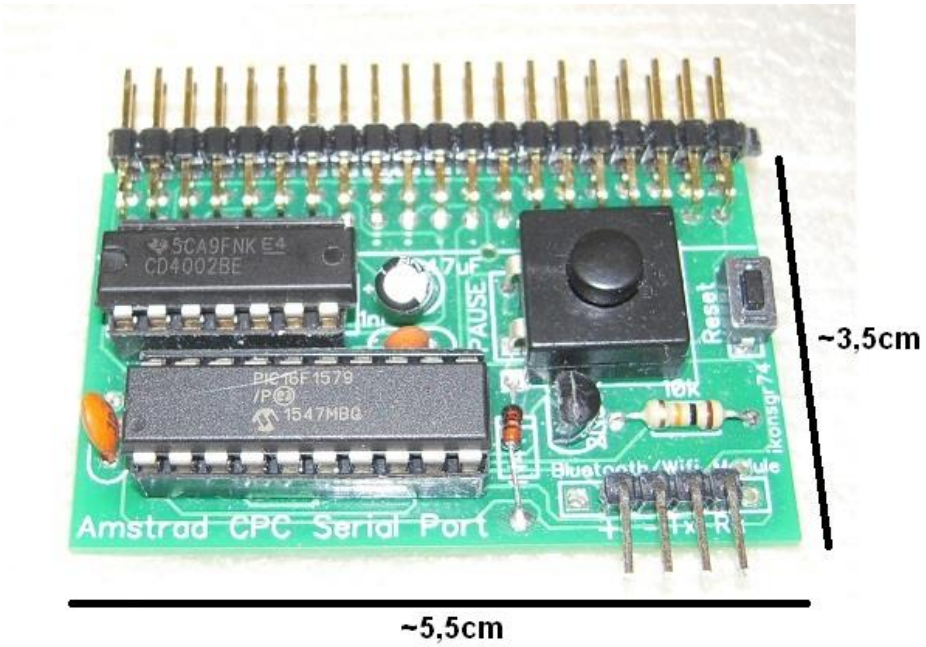
Amstrad CPC Serial Interface
User's Guide v1.2

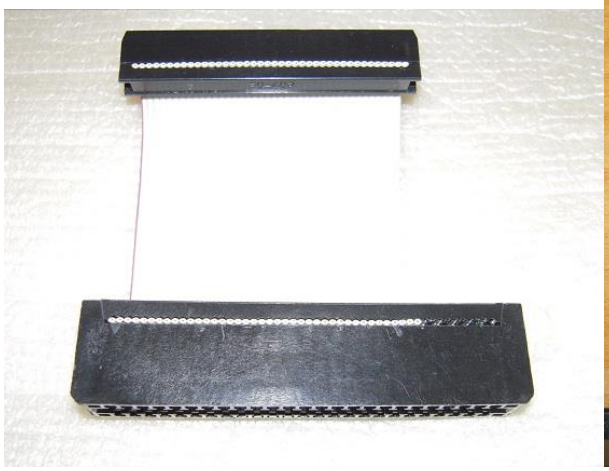
(John Konstantopoulos, Feb 2019)

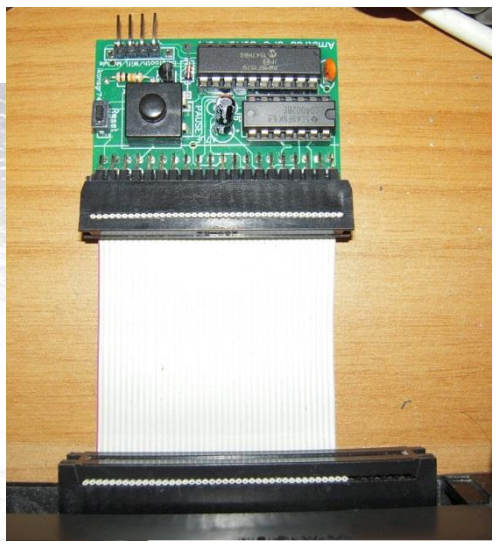
Introduction

Amstrad CPC serial interface is a universal serial port, based on the 8bit PIC 16F1579@32Mhz microcontroller. It can buffer up to 920 bytes of received data, and can achieve speeds up to 576000bps, using a usb2serial cable adapter, or 460800bps using a Bluetooth or Wifi Module. It is also equipped with a reset button and a Pause switch for Amstrad CPC:

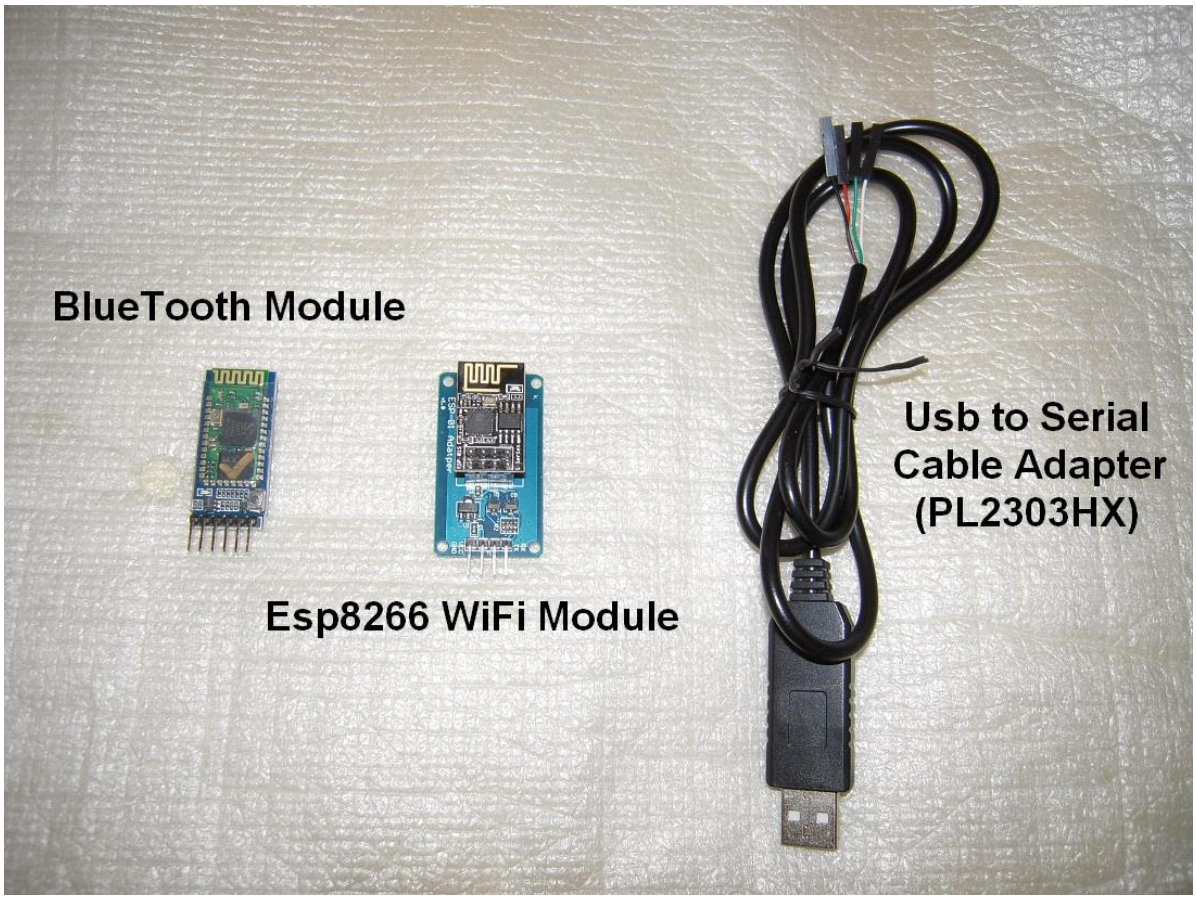


You can connect it to an external port raiser adapter like MX4 board, or directly to Amstrad's Expansion port using this adapter:





It allows serial connection through a usb2serial cable adapter, a wireless Bluetooth module or even a Wifi module:



All of these, are very easy to find on ebay and they cost almost nothing!(usb2serial can be found at ~1euro…).
Serial interface can be used with:

- A direct serial cable connection using only a usb2serial cable adapter.(Connect **ONLY** Tx, Rx, and ground to serial interface pins, red cable MUST NOT be connected to the 5V)
- A wireless connection using two Bluetooth modules (one connected to CPC serial interface, and the other to a usb2serial cable adapter),
- A wireless connection using an ESP-8266 wifi module!

BASIC Usage

Interface usage, is pretty simple and straight forward. Communication with the device is accomplished using only two ports:
&FBD1: The control port
&FBD0: The data port
And the two BASIC commands INP() for receive and OUT for send.

To send a byte, just give the BASIC command:
OUT &FBD0,x (x:0-255)
Or on assembler, the instructions:
LD A,x (x:0-255)
LD BC,&FBD0
OUT (C),A

To receive a byte, you must first check if there is any byte available at the buffer, by reading the control port. Use the BASIC command: **INP(&FBD1)**. If it returns '255', there is a byte available, if not, it returns '1'.So, in order to receive a byte from serial port, you just give the BASIC command:

IF INP(&FBD1)=255 THEN A=INP(&FBD0) (variable A contains the received byte)

And if you want to wait until a byte is available, you can use a small loop like:

```
WHILE INP(&FBD1)=1:WEND
A=INP(&FBD0)
```

And here is the assembly code for doing the same thing:

```
check_BYTE:
      LD A,&FB
      iN A,(&D1)
      DEC A
      JR Z,check_BYTE
ld      a ,&FB
in      a,(&D0)
```

Accumulator A now has the received byte.

Configure Interface

Default serial speed of the interface is 115200bps.
You can give various commands to the interface, using the control port, and the BASIC command:
OUT &FBD1,x
Where **x** is:

0: For resetting the interface (without resetting Amstrad CPC)

1: Clears receive data buffer

2: Enables a special “asynchronous burst” mode, for fast receiving the hole receive buffer (920 bytes) at once. When this mode is enabled, you don’t use INP(&FBD1) to check if there is available byte, but instead you receive directly data using only INP(&FBD0). When the receive buffer becomes empty, next INP(&FBD0) will cause automatically refilling of the receive buffer (e.g. 920bytes). This mode is used in the file transfer utility, in order to avoid asking if there is a byte available for every byte to receive. **USE WITH CAUTION,** because it will freeze Amstrad, until the receive buffer is full (920bytes)!

3: Disables the above mode

4: Will be used for Dsk image transfers (not yet implemented)

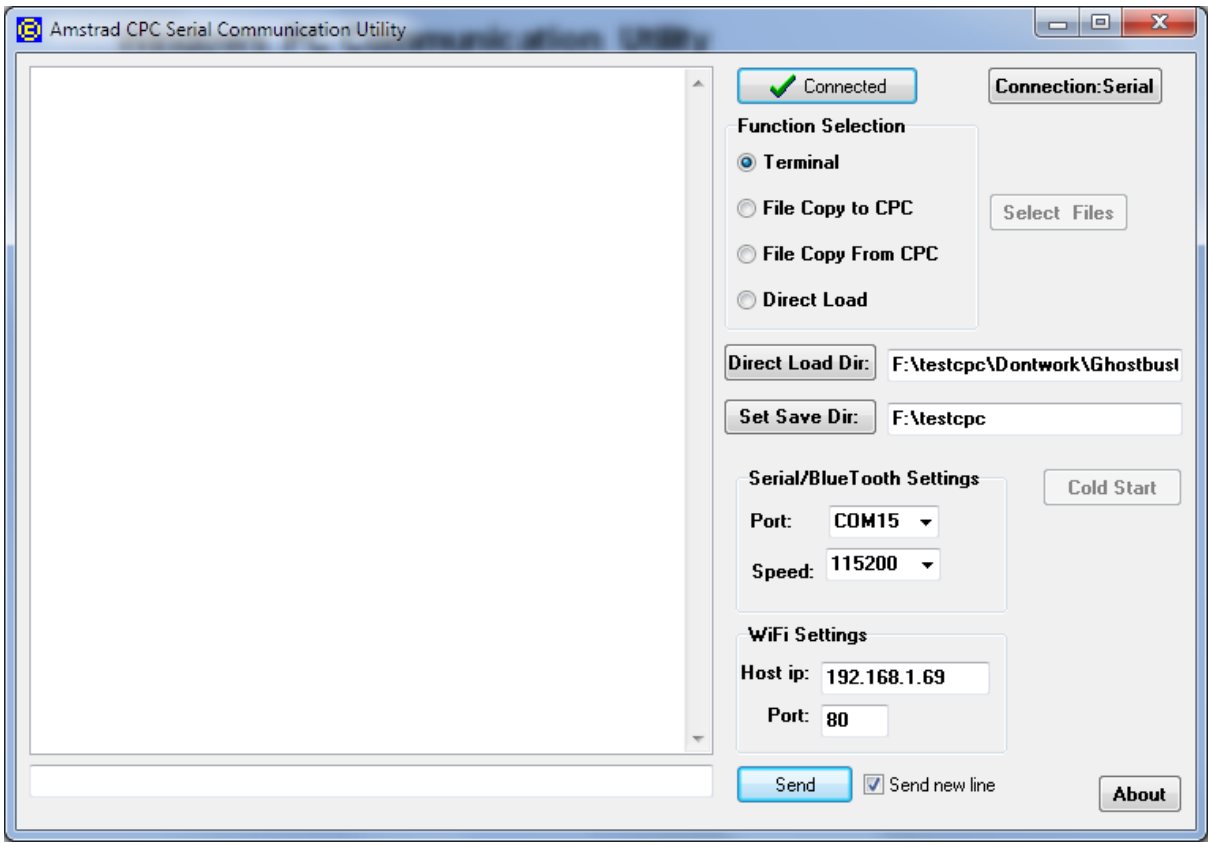
5: Enables the usage of Wifi module, needed ONLY when special “asynchronous burst” mode is used along with a WiFi module.

6: Disables the above

10: Set serial speed to 300bps
11: Set serial speed to 2400bps
12: Set serial speed to 9600bps
13: Set serial speed to 19200bps
14: Set serial speed to 38400bps (useful for setting Bluetooth module)
15: Set serial speed to 57600bps
16: Set serial speed to 115200bps (the usually default speed for Wifi Modules)
17: Set serial speed to 230400bps
18: Set serial speed to 460800bps (max speed using wireless Bluetooth/wifi modules)
18: Set serial speed to 576000bps (max speed using usb2serial direct cable adapter)
All commands for serial speed setting, return an acknowledge “SPEED:115200”, “SPEED:460800” etc.
30: Returns the status of interface, serial speed and if special Wifi mode is enabled

Windows PC Communication Utility

I have developed a utility for PC with windows, that allows communication with Amstrad CPC:



So far, it offers four basic functions:

- **Terminal:** A terminal for direct communication through serial/Bluetooth or Wifi modules.

- **File Copy to CPC**: A File Copy utility from PC directly to Amstrad CPC disk. You can select multiple files to transfer by pressing the “Select Files” button. Of course, you will have to run the corresponding utility on Amstrad CPC first (see next section “Serial Interface Utility Disk”).

- **File Copy From CPC:** A file Copy utility from Amstrad CPC disk to PC. Again,you need to run the corresponding utility on Amstrad CPC too.

- **Direct Load:** A method for Loading games/programs Directly from PC (see next section)

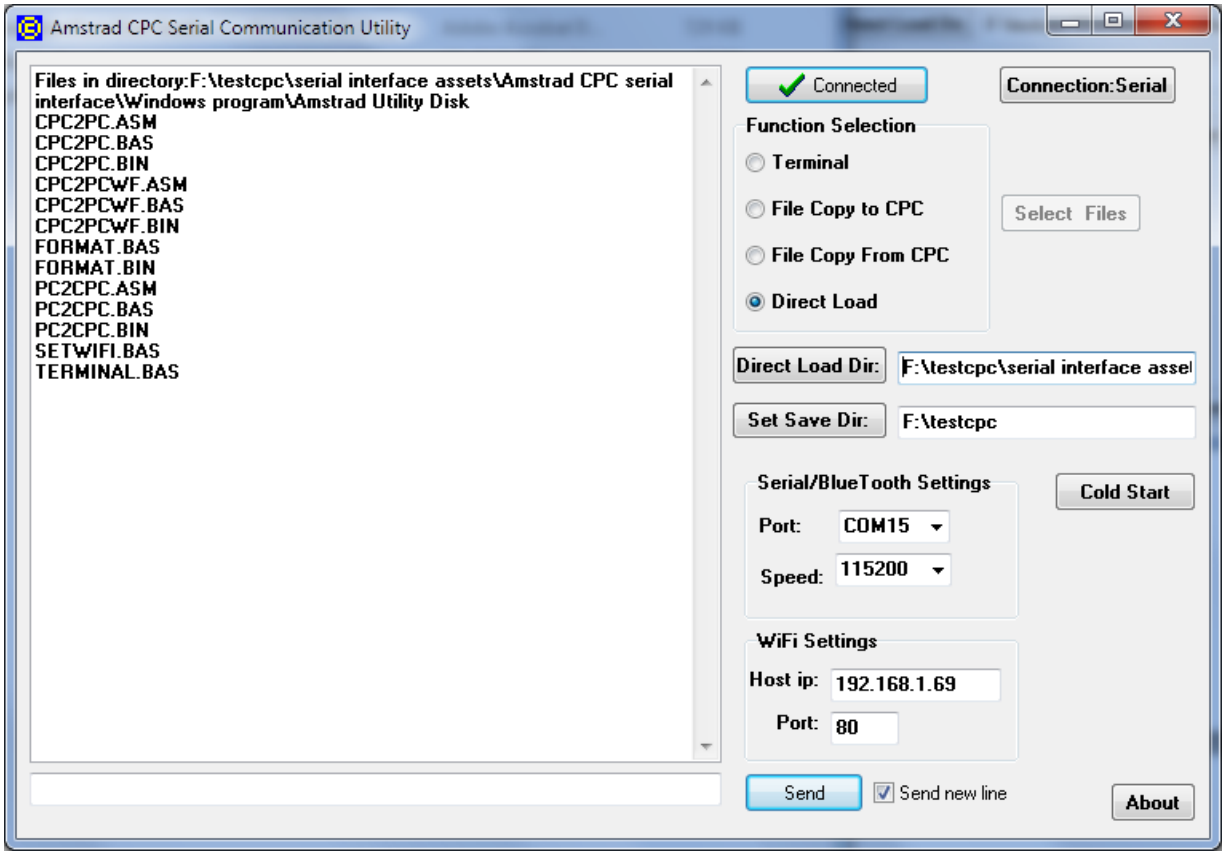
  Com port selection/speed for Serial connection and Host ip/Port for connection through WiFi, are also provided. **“Set Save Dir”** button,can be used for setting the saving directory for files copied from Amstrad CPC.

“Direct Load” Function

By selecting the Direct Load function, you will be able **to load or run a program (either BASIC or BINARY) and also many games, directly** from the PC!
The hole Procedure is rather easy and fast: First, run the windows application and set function to “Direct Load” (com speed will be automatically set to 115200). Then, type and save this small BASIC listing to your Amstrad CPC:

```
10 OUT &FBD1,0:OUT &FBD1,1:EI
20 FOR I=0 TO 341
30 POKE &A9B0+I,INP(&FBD0)
40 NEXT i
50 CALL &A9B0
```

Run the program, and after a couple of seconds, you will get the “Ready” prompt.
From now on, any LOAD” or RUN” command you will give, it will be redirected to PC!
Using button “**Direct Load Dir**”, you can set the loading directory and have a list of all available files in it:



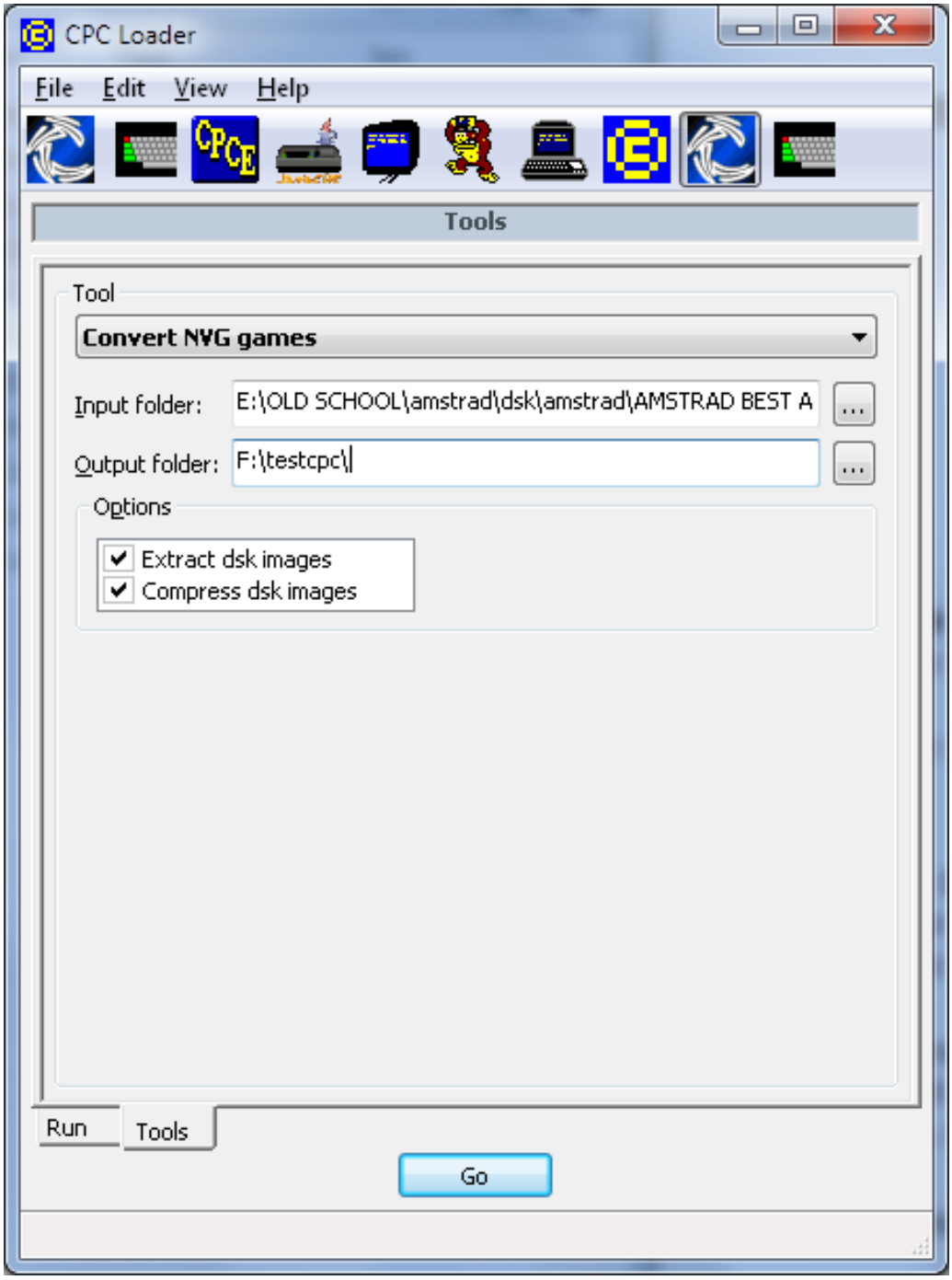
By setting loading directory to Amstrad Utility Disk folder, you can direct execute utilities **WITHOUT needing first to transfer the utility disk image to a real disk (or to a usb stick if you have a gotek/hxc drive)!**

For example, you can simply give:

RUN”Format.BAS” , in order to format a CPC disk, or

RUN”PC2CPC.BAS”, for transferring files from PC to Amstrad disks

After thorough testing, it is concluded that **~75% of games can be directly executed using this method!** You can find many dozens of games that already tested inside “direct load games” subfolder.
As this method requires to have files extracted from dsk images, i have also included the “cpc load” utility:



Using this, you can easily extract multiple dsk images instantly!
Finally, a couple of useful notes about Direct Load function usage:

- The small Basic program is needed to be executed, every time you want to load/run a new program/game. So it could be useful to save it on a disk or tape for easy reloading.

- After initial execution of BASIC program, you can try to reload the code (after resetting for loading another game/program), by just giving a: **CALL &A9B0** at BASIC prompt. This surely works for all BASIC programs and many simple single file games, and with some more mutli-file games too. If this doesn't work (Amstrad resets or crashes), you can reset Amstrad and run the small Basic program again.

- If you try to run/load a file that it doesn't exist, Amstrad will hung up. In this case, just reset Amstrad, and give **CALL &A9B0** at BASIC prompt.

Serial Interface Utility Disk

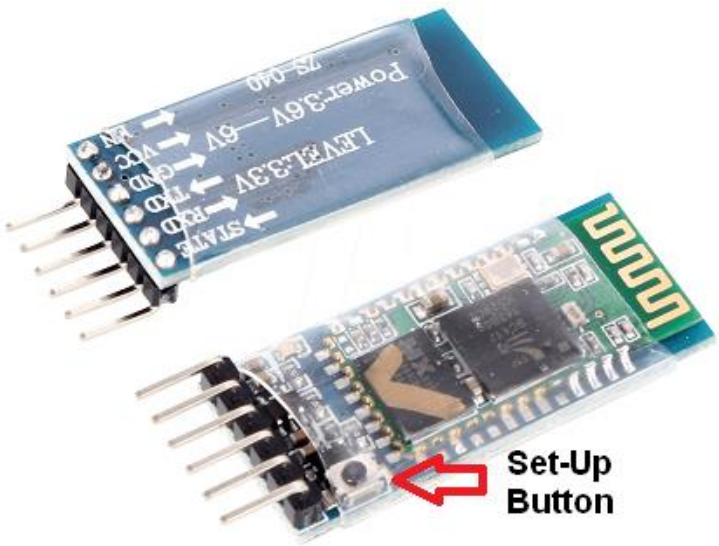
Along with the serial interface, a dsk image is also provided, containing many useful utilities:

- **Terminal.bas** A terminal program. Can be used for chat ,communication and to setup WiFi modules, Usb Host modules etc.
- **SETWIFI.BAS** A small utility to easy setup WiFi Module for use.
- **CPC2PC.BAS** A file transfer program to copy files from CPC disks directly to PC using a usb2serial cable or bluetooth module. Use arrows to move in catalogue, <space> to select files, <enter> to execute transfers, <1> to receive DRIVE A catalogue and <2> to receive DRIVE B catalogue.
- **CPC2PCWF.BAS** Like the above, but for using an ESP8266 WiFi Module instead.
- **PC2CPC.BAS** A file transfer program to copy files from PC directly to CPC disks.
- **FORMAT.BAS** A small format utility for easy formatting disks.

Many of the above programs, have also binary ".BIN" files containing the assembly routines for the various utilities, while ".ASM" files are ASCII text files, containing the assembly source code (can be directly used with Maxam assembler).
If you want to develop your own applications and programs, the ASM files are very useful for showing how to utilize the interface using assembly code, while the Terminal program, is a good example for BASIC programs.

Configure Bluetooth Module

The recommended Bluetooth module is HC-05:



You could also use the HC-06 which is a bit cheaper, but it lucks the Set-Up button, which is very useful, if you want to change the default speed of the module (9600bps). You can easily find these on ebay or electronics shops for a few euros.
Of course you will need a pair of these, to establish a bluetooth connection between Amstrad CPC and a PC or another Amstrad CPC. Or, you can use only one module for Amstrad CPC serial interface to connect with a tablet or a smart phone! ;-)
Connect Bluetooth module to the USB2Serial adapter cable (Red:5V, Black:Ground, Green:Tx-goes to Rx pin, White:Rx goes to Tx pin). Then, connect cable's usb plug to a free usb port of your PC, **WHILE pressing the setup button**. The red led should flash every 2-3 seconds. Now, you can easily configure Bluetooth modules using a terminal program or even the PC communication utility I have developed.

First, set speed to 38400 and give the commands:
AT -You should get respond "OK"
AT+UART=X,0,0
X, can be 2400,9600,19200,38400,57600,115200,230400,460800. You should get response "OK".
Now, disconnect the Bluetooth module and connect the other one (don't forget to press the button while powering to enter the setup mode).Give the above commands, but also add this one:
AT+ROLE=1 –You should get response "OK"
This is for making the module "Master". In order for two Bluetooth modules to connect to each other, one must be "Master" and the other "Slave". But Since all BT modules are preconfigured as "Slave", you should change one of them to "Master".
Finally, power off and on the BT module (in order to exit from setup mode) and connect the other BT module to serial interface. After a few seconds, **both modules should flash twice every 3-4 seconds**. This means that they are connected, and you are ready to go!

Configure ESP8266 WiFi Module

You can directly connect an ESP8266 WiFi module to serial interface.
You can either use:

Esp-07 **or** **Esp-01/01S**

The latter, will also need this 5v adapter (ESP-01 adapter):

Which will be combined to this:

You can easily find any of these, on ebay or many electronics shops, at very cheap prices, of only a few euros. And remember to connect Tx to serial's interface Tx pin, and Rx to Rx pin (no swap of signals needed).

Now, in order to use it , module must be connected with an Access point and start a small TCP server on it. This can be done easily by using the small utility "SetWiFi" provided on Amstrad CPC Serial Interface utility disk. First select the speed 115200 (which is usually the default speed of Wifi module) and then, the new desired speed (up to 460800). Then, you just give the name and password of the access point (where WiFi module will be connected to),and if everything goes well, program will return you the given ip address to module. This ip address is used on the PC utility program to establish connection between Amstrad CPC and the PC.

ESP8266 module use AT commands for control and communication. There is a very big manual with 100's of these available, but bellow, i give you only the few commands you will ever need, for configuring the module manually, using the terminal program:

AT+CWLAP will list access points in range
AT+CWJAP="yourSSID","yourWifiPassword" connects to an access point
AT+CWQAP quit from acess point
AT+UART_CUR=115200,8,1,0,0 (don't remain after reboot/power off)
AT+UART_DEF=115200,8,1,0,0 Set UART speed (remains after reboot/power off)
AT+CWMODE = 1 (for ST mode , 2 for AccessPoint mode & 3 for both)
AT+CIPMUX=1 (set multiple connections, this must be set to '1')
AT+CIPSERVER=1,80 (start the TCP server at port 80)
AT+CIFSR (shows WiFi module's ip/mac address)
ATE0 remove echo (must be set in order to use file transfer utility)
AT+CIPSEND=0,x (send x bytes. Bytes/chars follow immediately after the command)
AT+CIPSTO=X (server timeout, range between 0 and 7200, set to large number or 0 for never)
AT+CIPSTART="TCP","Tcp server IP",port no. Connects to a TCP server, useful for connecting many Amstrad CPC's together in a WiFi network
AT+CWSAP_DEF=<ssid>,<pwd>,<chl>,<ecn> Set up WiFi module's Access point
<ssid>: string parameter, SSID of AP.
<pwd>: string parameter, length of password: 8 ~ 64bytes ASCII.
<chl>: channel ID.
<ecn>: encryption method; WEP is not supported.

- **0: OPEN**
- **2: WPA_PSK**
- **3: WPA2_PSK**
- **4: WPA_WPA2_PSK**

DO IT YOURSELF

You can find everything you need to make the adapter yourself, here. It contains, the circuit diagram,the bill of materials, the gerber files for the pcb, the Pic 16F1579 microcontroller Hex file (you will need a pickit2/3 in order to flash it into PIC 16F1579), Amstrad's utility dsk image disk (you can transfer it to a real disk using CPCDISKXP),and of course the PC utility program.