

# **Amstrad CPC Serial Interface (a.k.a USIfAC)**

## **User's Guide v1.5**

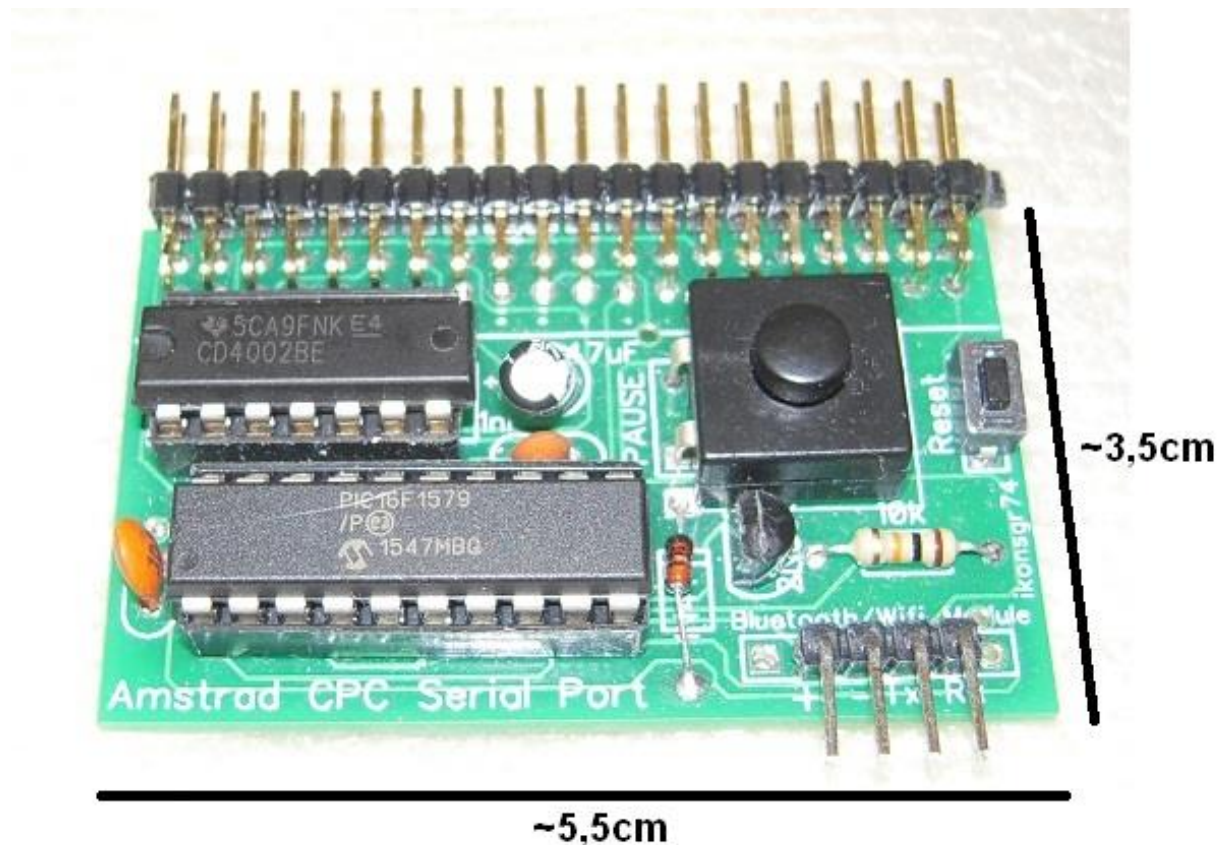
(John Konstantopoulos, July 2019)

### **Table of contents**

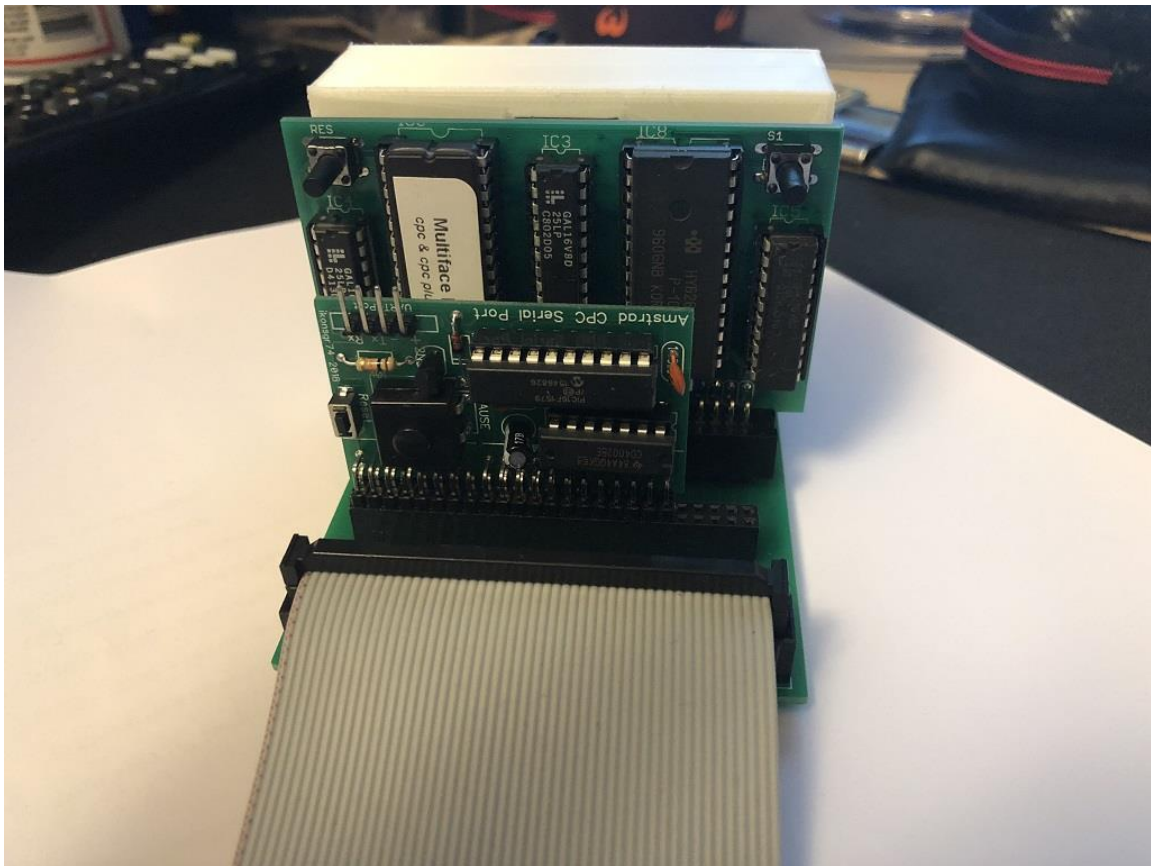
<b>Introduction.....</b>	<b>2</b>
<b>Basic Usage .....</b>	<b>5</b>
Send & Receive bytes: .....	5
Send & Receive bytes with a WiFi module acting as TCP server: .....	6
<b>Configure Interface .....</b>	<b>7</b>
<b>Windows PC Communication Utility.....</b>	<b>8</b>
"Terminal" Function .....	10
"File Copy to CPC" Function .....	12
"File Copy From CPC" Function .....	14
"Image Copy" Function .....	15
"Direct Load" Function .....	17
<b>Serial Interface Utility Disk.....</b>	<b>19</b>
<b>Configure Bluetooth Module.....</b>	<b>20</b>
<b>Configure ESP8266 WiFi Module .....</b>	<b>21</b>
<b>Do it Yourself .....</b>	<b>23</b>

## Introduction

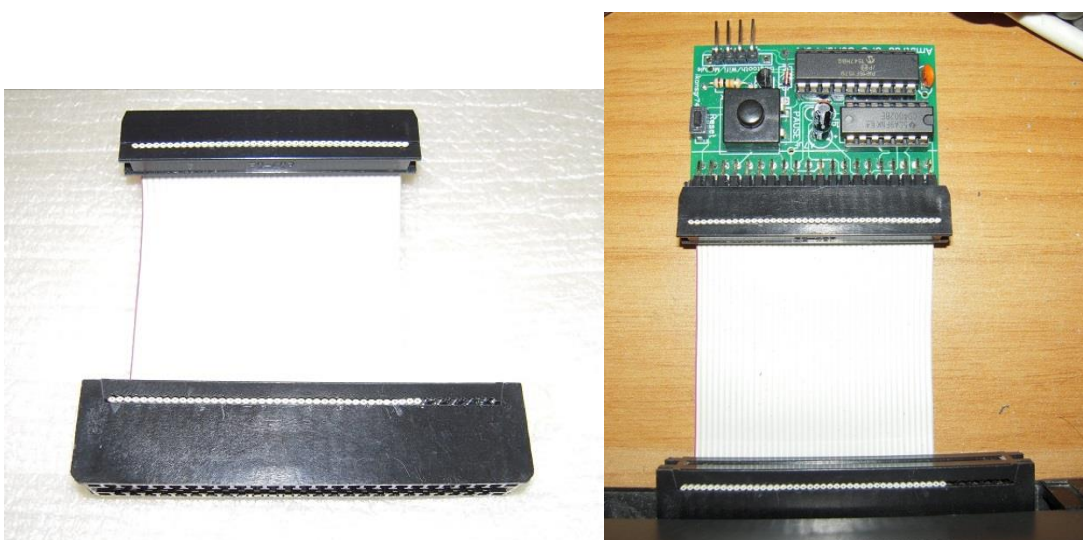
Amstrad CPC serial interface is a universal serial port, based on the 8bit PIC 16F1579@32Mhz microcontroller. It can buffer up to 920 bytes of received data, and can achieve speeds up to 576000bps, using a usb2serial cable adapter, or 460800bps using a Bluetooth or Wifi Module. It is also equipped with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



You can connect it directly to an external port raiser adapter like MX4 board:

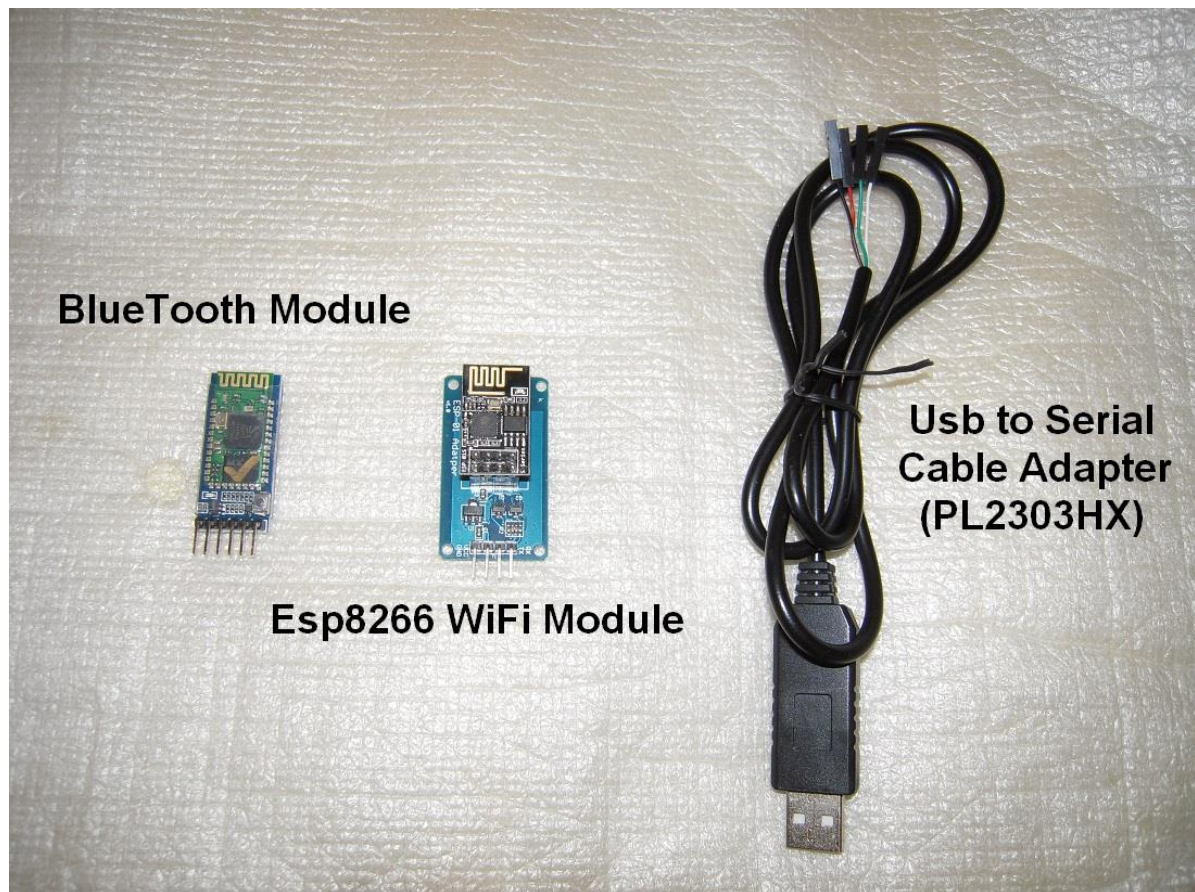


Or directly to Amstrad's Expansion port using this adapter:





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 a usb2serial cable adapter.(Connect **ONLY** Tx=Green, Rx=White, and Ground=black to serial interface pins, red cable **MUST NOT** be connected to the 5V=Red)
- A **wireless connection using a Bluetooth module** for direct wireless connection to a Laptop/Note book/Smart phone equipped with Bluetooth, or, two Bluetooth modules, one connected to CPC serial interface, and the other to a usb2serial cable adapter connected to a usb port.
- A **wireless connection using** an ESP-8266 wifi module! This is especially useful for creating a LAN or for direct connection to internet!
- If you have an on old motherboard with **on board RS232 serial port**, you can't use it directly to connect with serial interface, but you must use an [RS232 to TTL converter](#) between them.

## 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.

### Send & Receive bytes:

This method can be used with direct serial/ Bluetooth module and also, WiFi module, which must be pre-configured using the “**Setwifi**” program, in order to establish a passthrough/transparent transmission mode.

**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.

## **Send & Receive bytes with a WiFi module acting as TCP server:**

In order to **send data using a WiFi module** that is configured as TCP server (using the “**setservr**” program), you must first execute an AT command:

**AT+CIPSEND=0,X**

Where X, is the length of the string to be sent. Data string follows immediately after.  
Below is the BASIC routine **to send a string of chars/bytes (named A\$)**:

```
1000 REM SEND WiFi ROUTINE
1010 B$=STR$(LEN(A$)):C$="AT+CIPSEND=0,"+RIGHT$(B$,(LEN(b$)-1))
1020 FOR k=1 TO LEN(c$)
1030 OUT &FBD0,ASC(MID$(c$,K,1))
1040 NEXT K
1050 OUT &FBD0,&D:OUT &FBD0,&A
1060 FOR k=1 TO LEN(a$)
1070 OUT &FBD0,ASC(MID$(A$,K,1))
1080 NEXT K
1090 RETURN
```

**Receiving a byte/char**, is done like with serial connection, BUT before getting the actual data, there is always a header in the form of:

**+IPD,1,X:**

Where **X** is the number of the received bytes (or length of string).

In order to **receive only the actual data**, some starting bytes should be removed:

For **1-9 bytes/chars**, first **13 bytes** should be removed or ignored

For **10-99 bytes/chars**, first **14 bytes** should be removed or ignored

For **100-999 bytes/chars**, first **15 bytes** should be removed or ignored

For **1000-2048 bytes/chars**, first **16 bytes** should be removed or ignored

## 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 whole 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

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)

19: Set serial speed to 576000bps (max speed using usb2serial direct cable adapter)

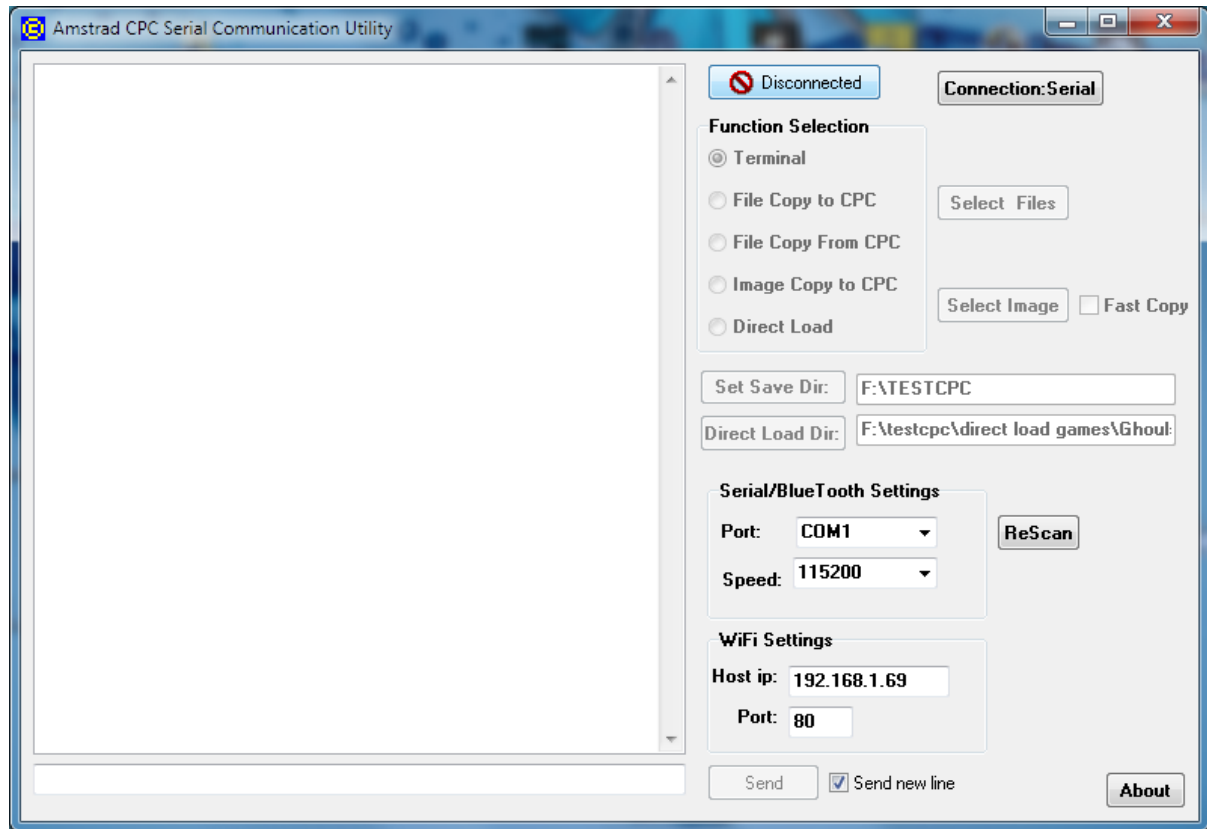
20: Set serial speed to 31250bps useful for connecting MIDI devices

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:



It offers five 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.
- **File Copy From CPC:** A file Copy utility from Amstrad CPC disk to PC.
- **Image Copy to CPC:** A disk image copy utility from PC to Amstrad CPC disks.
- **Direct Load:** A method for Loading games/programs Directly from PC

For more details on the above functions, see next sections.

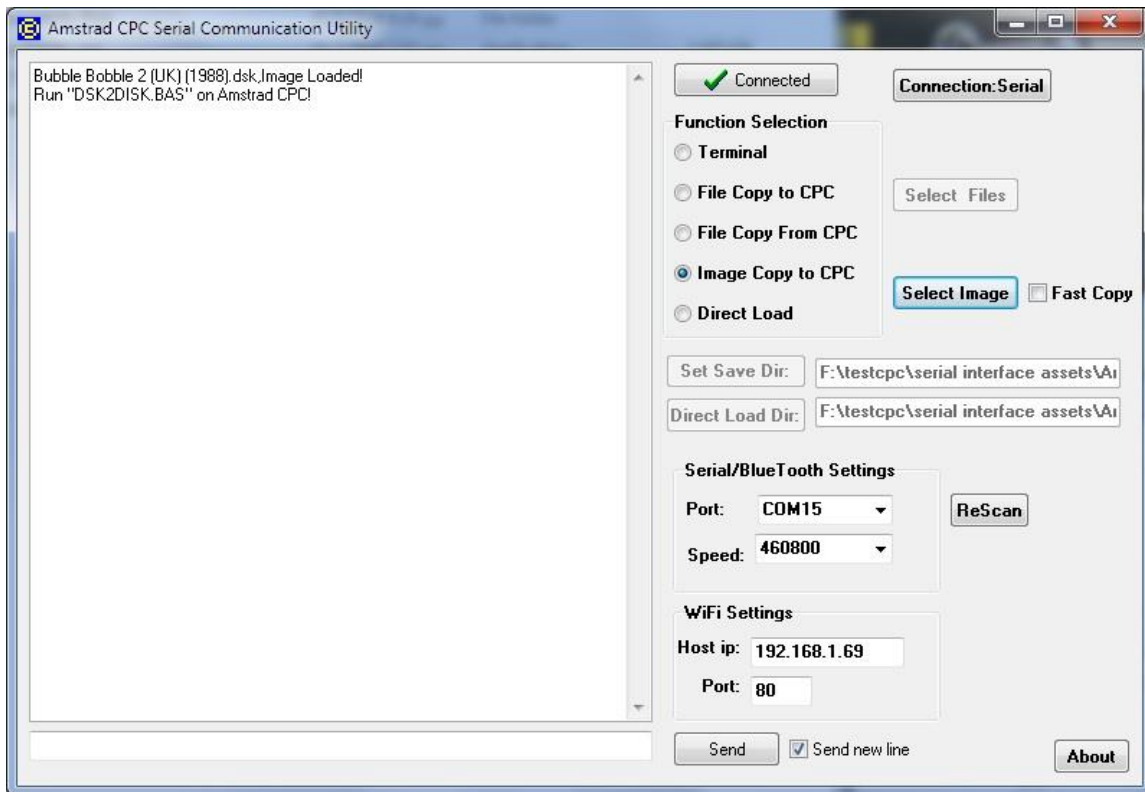
Now, in order to connect with serial interface on Amstrad CPC, you must first setup your connection. This needs to be done only the first time, program will “remember” your settings, and set them automatically every time you run the utility.

First, using the **connection** button on the right top, select the type of your connection (Serial or WiFi), according to what you are using.

Then, from groupbox “**Serial/Bluetooth Settings**” you can select Com port (note: “**Rescan**” button, allows you to rescan for serial com ports, in case you enable/disable Com ports after running the utility), and speed for Serial/Bluetooth connection. If you use a WiFi module, connection is established from AMSTRAD CPC (see section: “configure Esp 8266 WiFi module”). Make sure that Serial interface on Amstrad CPC has the same



speed settings and, if you are using Bluetooth module, the module is properly set (see “Configure Bluetooth Module” section, on how exactly to do that).  
Press **Disconnected** button, it should turn to **Connected**:



Following bellow, are details on using each function.

## “Terminal” Function

On Amstrad CPC, run “Terminal.BAS” program. First set Serial (for cable or Bluetooth) or Wifi module usage:



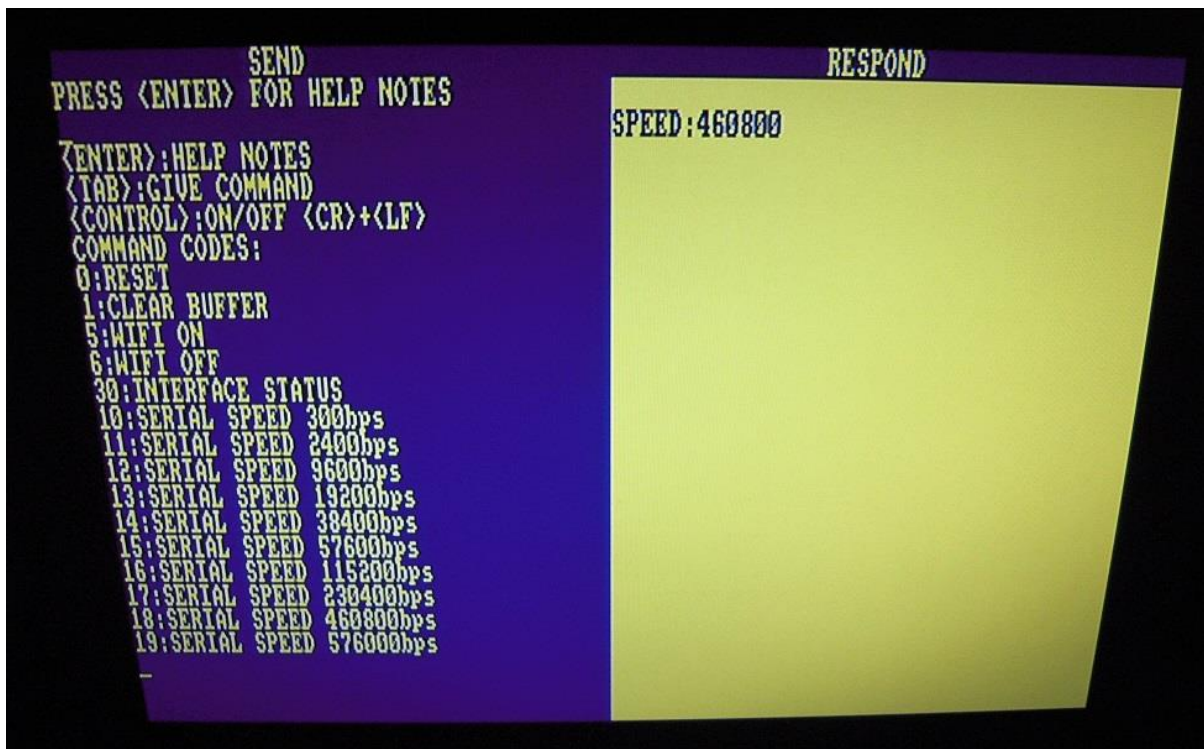
And then set serial port speed:



If you have selected WiFi mode and the module is not connected, you will be asked to give Access point's ssid/password (this will be omitted if module is already connected to

an A.P.), and then TCP server's ip and port (these are obtained from groupbox **"WiFi Settings"**, in PC utility program window.

After that, the main terminal window appears:



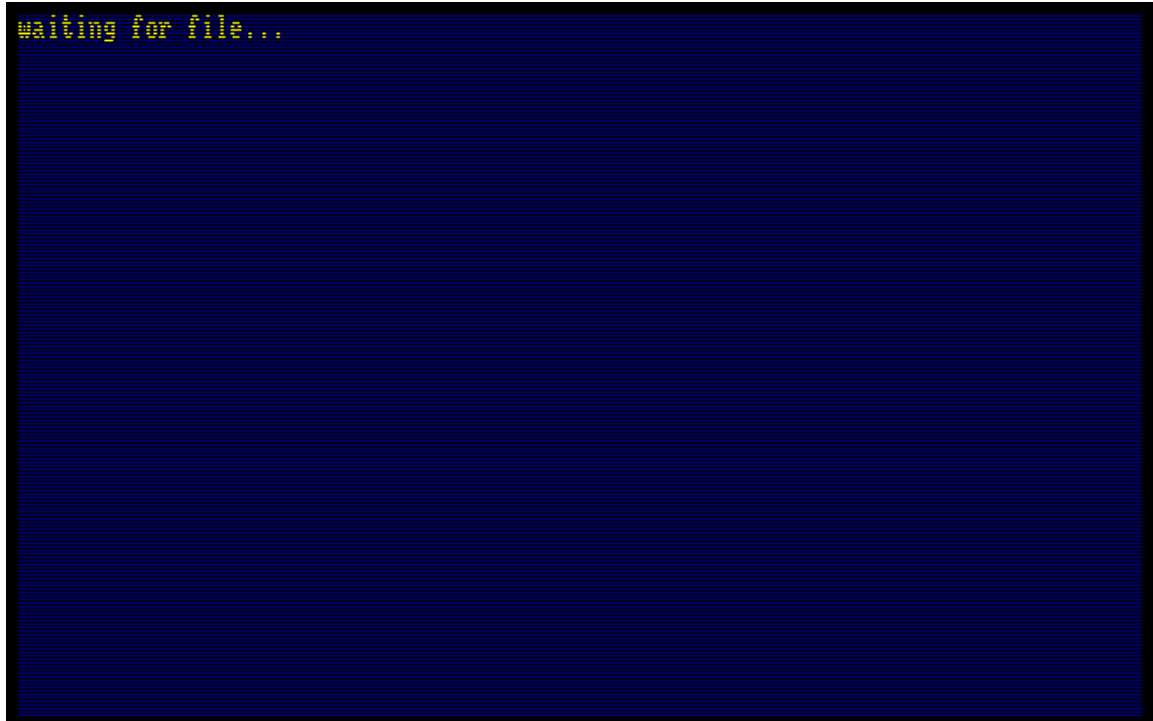
On the left side, write anything, and by pressing <Return> it will be transmitted to PC. On the right side you will receive any respond from the PC.

With terminal, you can also communicate with serial interface by sending various commands to change port speed, clear receive buffer, get status etc. By pressing <Enter> (or <COPY> on WiFi mode), you will get a list of all available commands. Commands are sent by pressing <Tab> and then the desired code.

On PC side, after setting and enabling connection, select Terminal function. You can use the edit box on the left of the <Send> button to write a message and by pressing Send button it will be transmitted to Amstrad CPC.

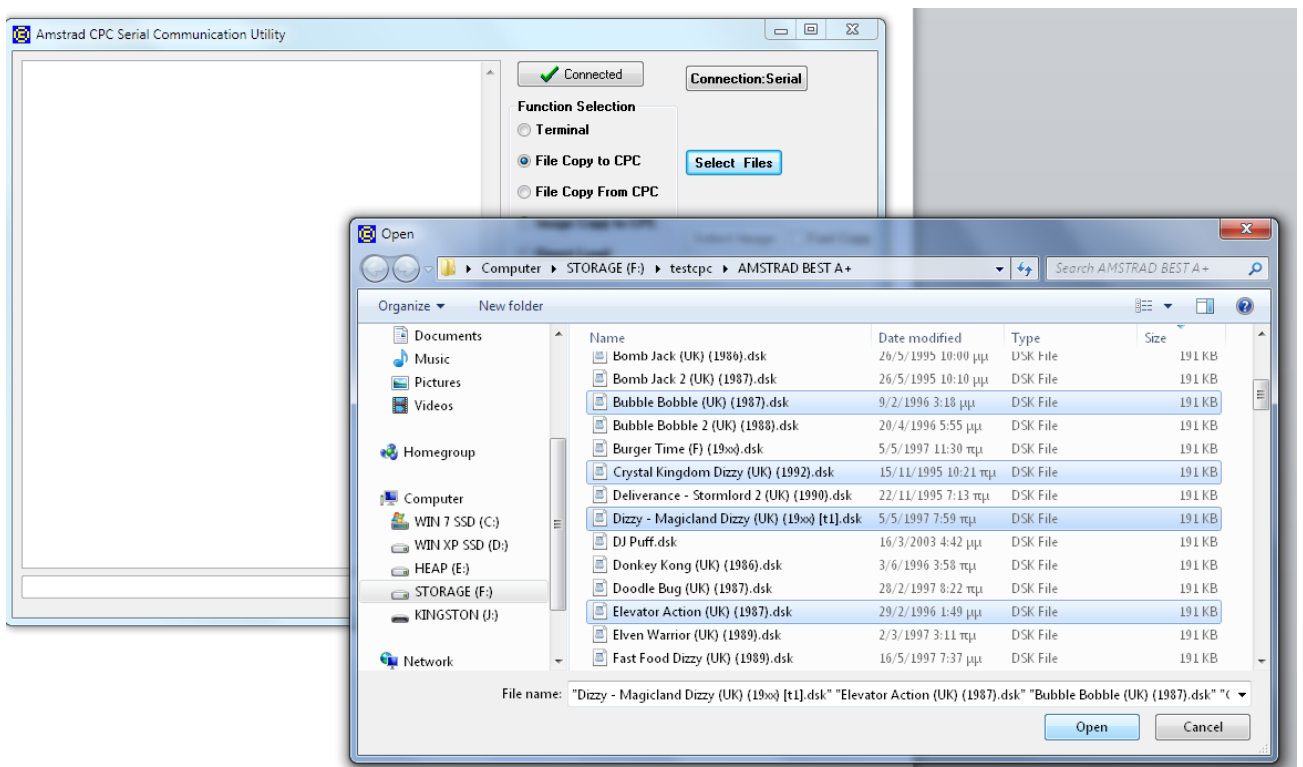
## “File Copy to CPC” Function

On Amstrad CPC, run “PC2CPC.BAS”. You need to set the serial port speed (for WiFi mode, don’t forget to run first “SetWifi” program to establish the connection) , and then, you should get a “waiting for file...” message on screen:

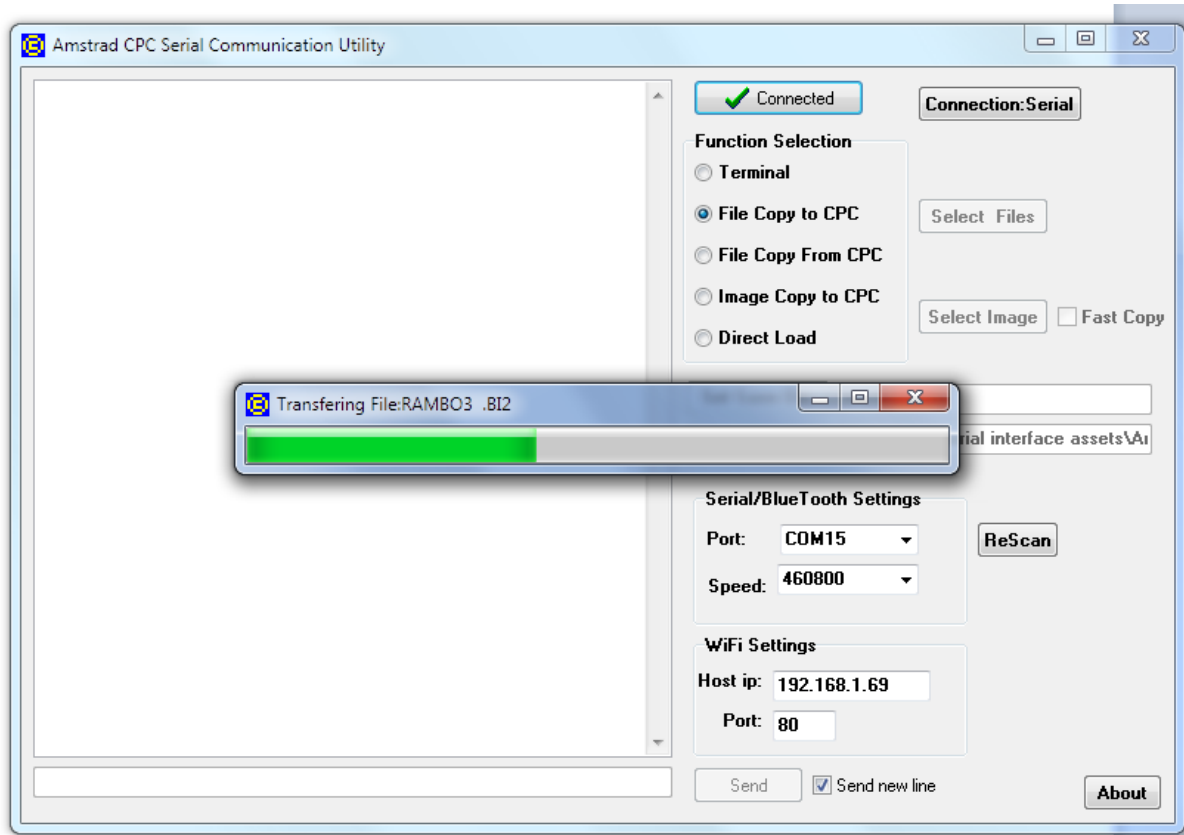


Insert a formatted disk into Disk drive.

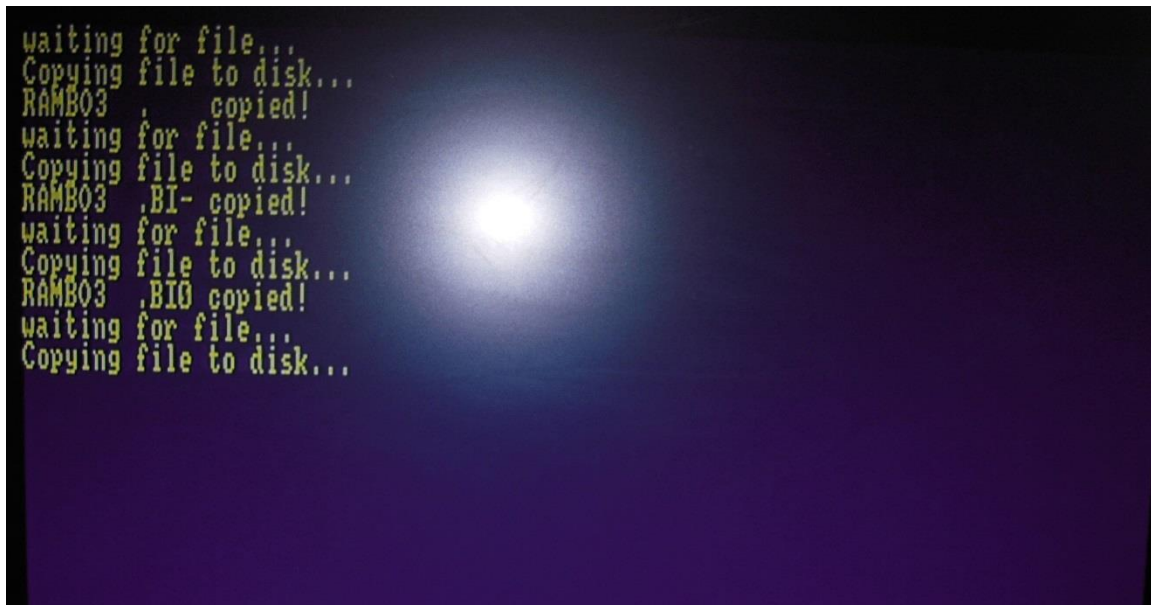
Run windows utility on the PC. After setting and enabling connection, select “File Copy to CPC” function. Then, press “Select Files” button and select one or more files to transfer:



Press “Open” Button and file copy will begin:



On Amstrad, you will get a confirmation for each written file:



After finishing transferring the files you want, you can reset Amstrad to exit the program.

## “File Copy From CPC” Function

On PC, after setting and enabling connection, select “**File copy From CPC**” function. Use the “**Set Save Dir**” button, to set the save directory for the files transferred from CPC. On Amstrad CPC, run “**CPC2PC.BAS**. After setting serial speed, you will get a files catalogue of the inserted floppy disk:

```
Drive A: user 0

DSK2DISK.BIN  MONITOR.BIN  MONITOR.BAS  PC2CPC.ASM  PC2CPC.BAS
PC2CPC.BIN  SETWIFI.BAS  TERMINAL.BAS  CPC2PC.ASM  DIRECT.BAS
CPC2PC.BIN  CPC2PCWF.ASM  CPC2PCWF.BAS  CPC2PCWF.BIN  DSK2DISK.BAS
CPC2PC.BAS

121K free

Files to transfer:PC2CPC .BIN  FORMAT .BAS  DSK2DISK.BIN  PC2CPC .ASM
                  SETWIFI .BAS  FORMAT .BIN  PC2CPC .BAS  DIRECT .BAS
```

Use **arrows** to move between files, **<space>** to select/unselect a file, and **<Return>** to execute transfers of the selected files. By pressing **<1>**, you will receive DRIVE A file catalogue and by **<2>**, DRIVE B file catalogue.

Finally, with **<Caps Lock>** you can select/unselect all the files. This is useful if you want to transfer all files of the disk, or if you want to “invert” the already selected files (e.g. all selected files will be unselected, and all other, will be selected).

On the half bottom of the screen, you will get a list of all the selected files. Files from this list, will be removed one by one, during the transfer process:

```
Drive A: user 0

DSK2DISK.BIN  MONITOR.BIN  FORMAT .BAS  PC2CPC.ASM  PC2CPC.BIN
PC2CPC.BIN  SETWIFI.BAS  TERMINAL.BAS  CPC2PC.ASM  DIRECT.BAS
CPC2PC.BIN  CPC2PCWF.ASM  CPC2PCWF.BAS  CPC2PCWF.BIN  DSK2DISK.BAS
CPC2PC.BAS

121K free

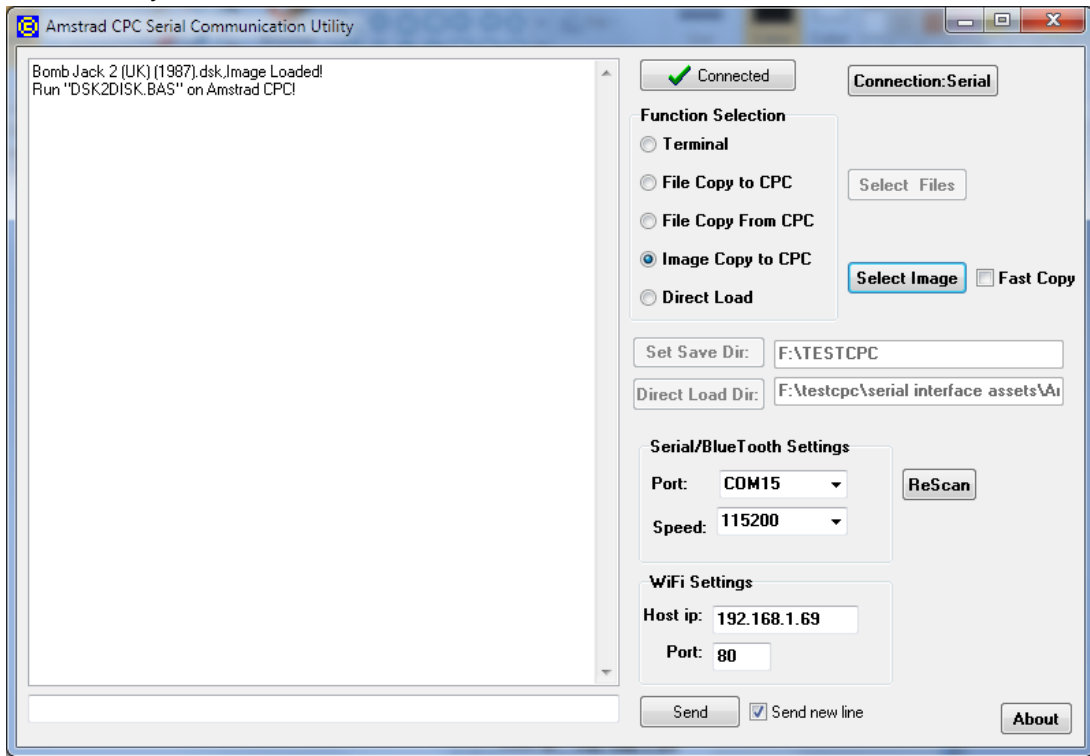
Files to transfer:DIRECT .BAS  PC2CPC .BAS  DSK2DISK.BIN  PC2CPC .ASM
                  SETWIFI .BAS  FORMAT .BIN

Transferring DSK2DISK.BIN...
```

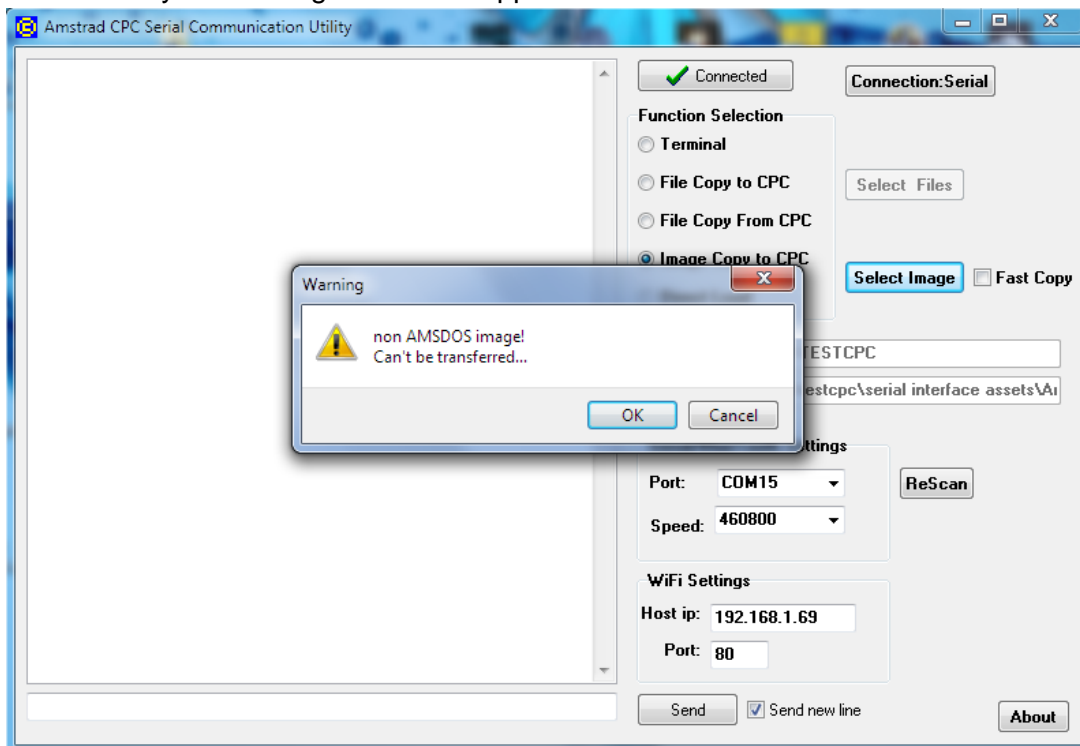


## “Image Copy” Function

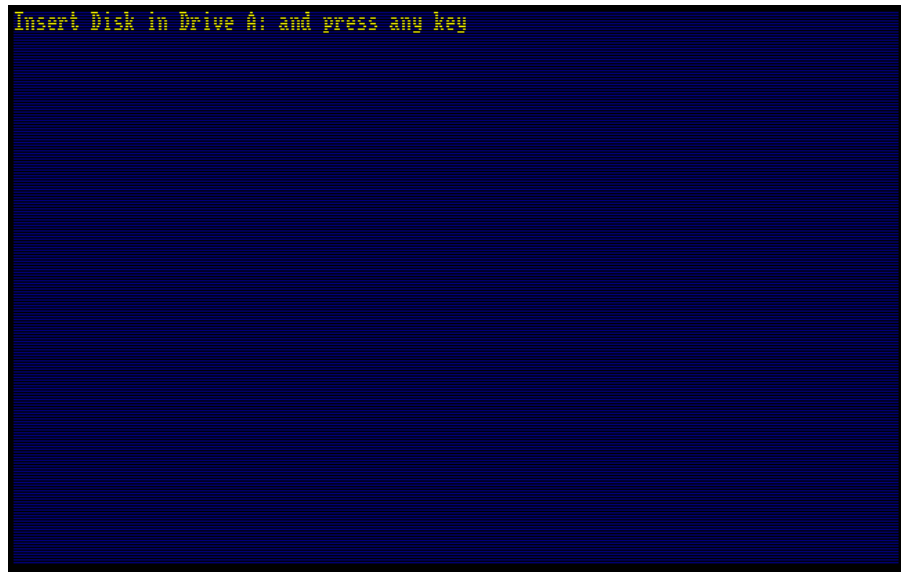
On PC utility program, select **“Image copy to CPC”** function. Press **“Select Image”** button to choose a disk image. You will get a confirmation that image is loaded successfully:



If you select a non Amsdos image (usually copy protected games) you will be notified that currently these images are not supported:

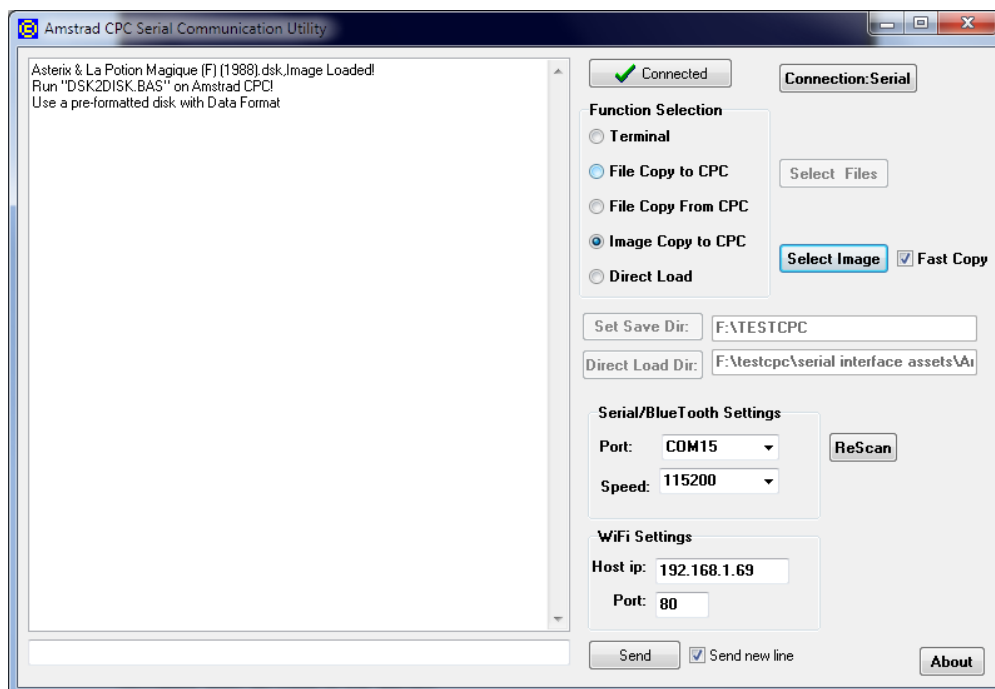


Now, run the “DSK2DISK.BAS” program on the CPC, and after setting serial speed, you will get a message to insert a disk:



By pressing any key, the transfer of the selected image will begin. Using serial speed of 460800bps or more, it should take less than a minute (~55-56seconds) to transfer a hole 180Kb disk image into any disk (either CPC/ PC/Apple formatted or unformatted).

Now, as you might already noticed, there is also a “**Fast Copy**” check box, right next to Select image button. By checking this, **you can transfer an image to an already Amstrad Formatted disk in just ~40 seconds!** If you check “Fast Copy”, you will be notified, for the format type needed (in most cases is Data Format) on the disk:



This can be proved very useful if you already transfer a game image to a disk, and you want to transfer another image on the same disk.

## “Direct Load” Function

(This function works only in serial/Bluetooth mode)

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 whole Procedure is rather easy and fast:

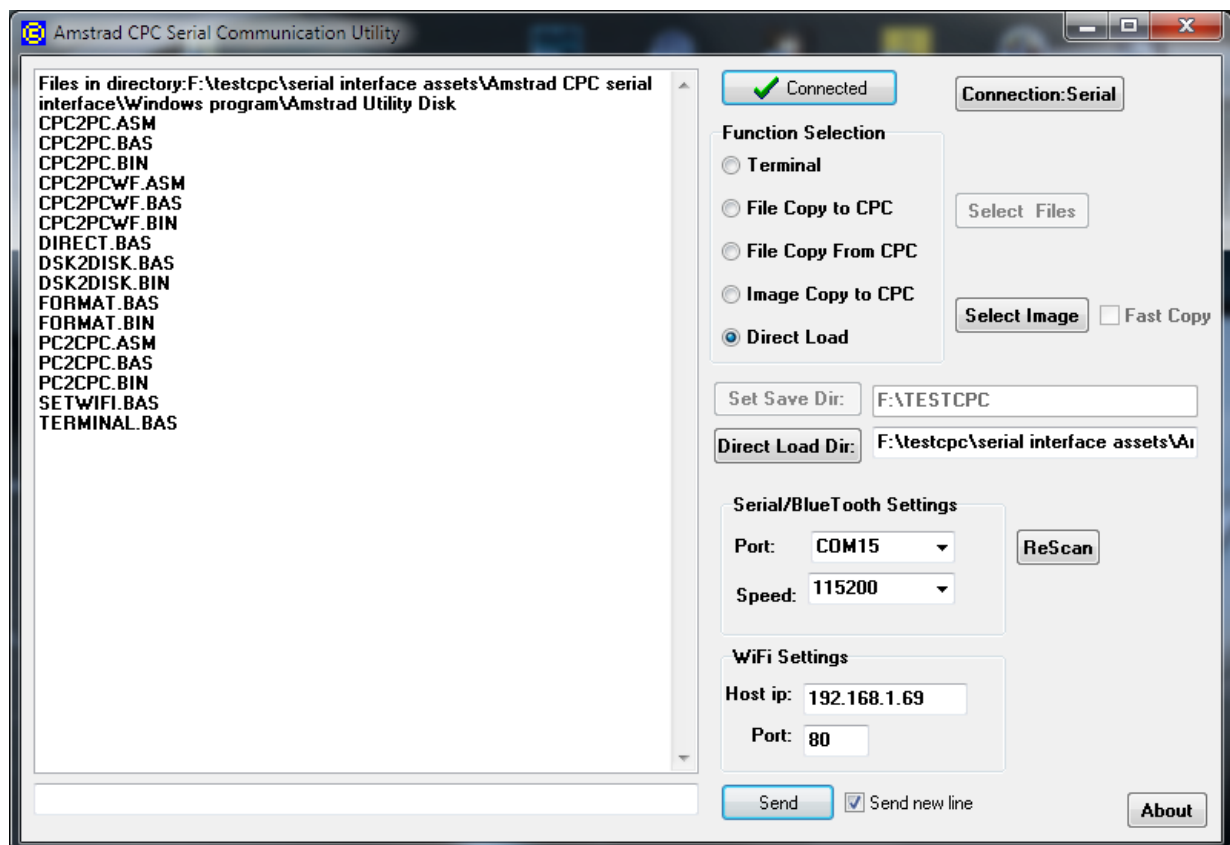
First, run the windows application and set function to “Direct Load” (com speed will be automatically set to 115200). Then, type and save this small BASIC listing to your Amstrad CPC:

```
10 OUT &FBD1,0:OUT &FBD0,0:DI:DI
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:

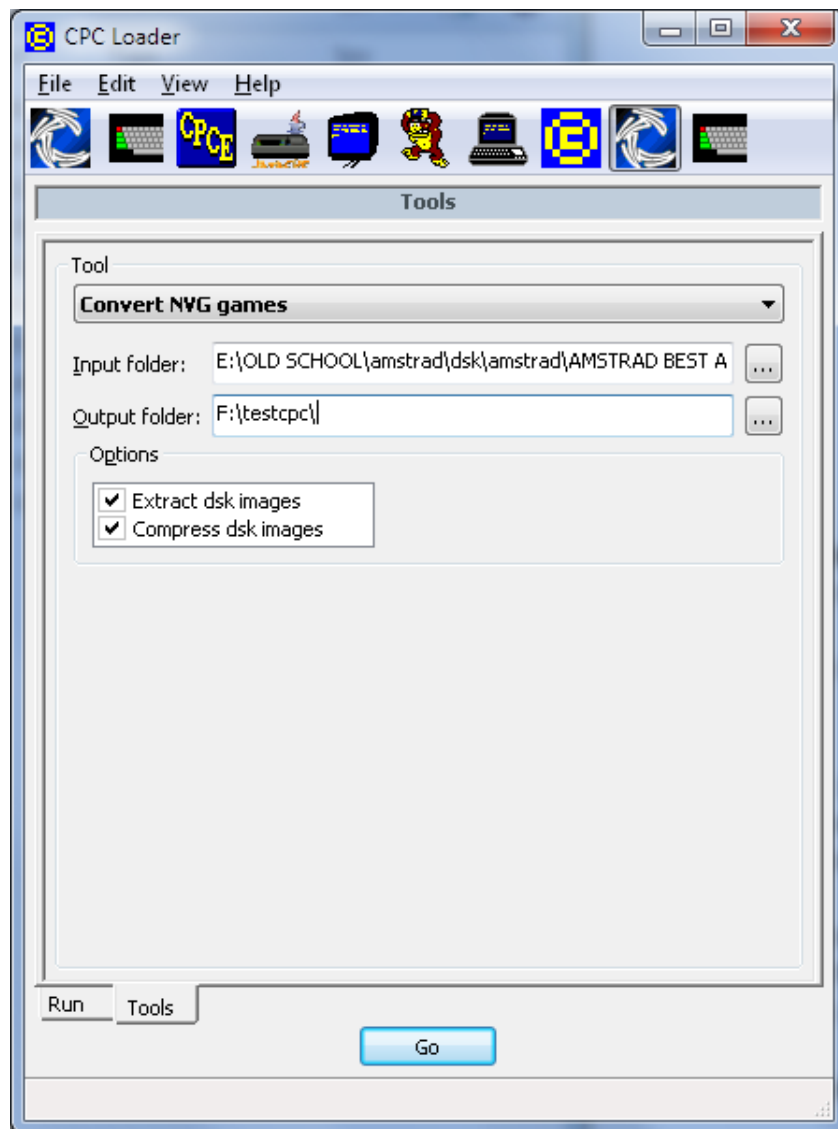
**RUN"Format.BAS"** , in order to format a CPC disk,

**RUN"PC2CPC.BAS"**, for transferring files from PC to Amstrad disks (after program loads, you should select "File Copy to CPC" function)

**RUN"DSK2DISK.BAS"**, for transferring dsk images from PC to Amstrad disks (after program loads, you should select "Image Copy to CPC" function)

Only the "File Copy from CPC" function can't be "direct" executed, this, you must first copy it to a CPC disk.

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 [here](#). As this method requires to have files extracted from dsk images, i have also included the "cpc loader" 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 also with some 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 hang. In this case, just reset Amstrad, and give **CALL &A9B0** at BASIC prompt.
- For loading files with ".BIN" extension, this can be omitted (e.g. you can type:**RUN"GAME"**, instead of:**RUN"GAME.BIN"**, regardless of the file having the "BIN" extension or not). But you must include extension to load any ".BAS" files.
- If, after initial running of the small basic program, you have any problem loading a game/program, try to turn off/on Amstrad, and if this don't work, go to PC program and select momentarily another function (like "terminal") and then set to "Direct Load" again (this will reinitialize the PC routine in case it has hang/crash).

## Serial Interface Utility Disk

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

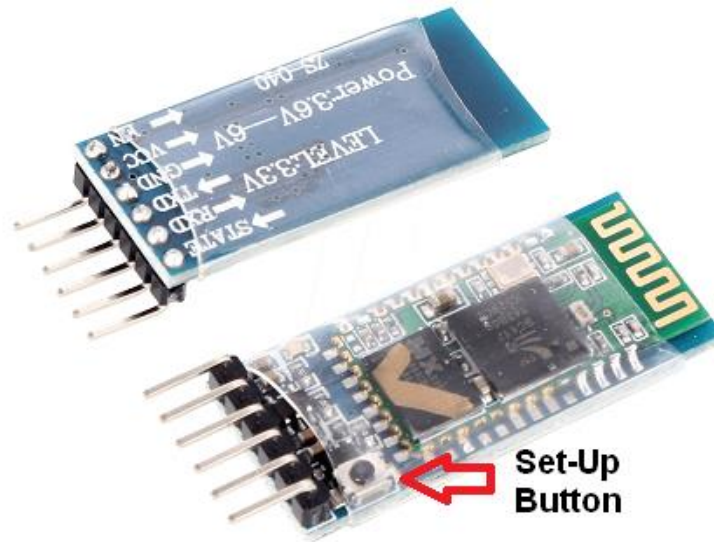
- **Terminal.bas** A terminal program. Can be used for chat ,communication and to setup WiFi modules, Usb Host modules etc.
- **SETWIFI.BAS** A small utility to easy setup WiFi Module for use.
- **CPC2PC.BAS** A file transfer program to copy files from CPC disks directly to PC using a usb2serial cable or bluetooth module.
- **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.
- **DSK2DISK.BAS** A utility to transfer DSK images from PC to disks.
- **SETSEVR.BAS** A utility for easy setup a TCP server on WiFi module, in order to create an Amstrad CPC LAN network!

Some 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 lacks 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 WiFi, module must be connected to an Access point and then connect to the TCP server that is activated by the PC utility program. This can be done either using terminal utility to give the appropriate AT commands, or more easily, by using the small utility program "**SetWiFi.BAS**" 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 then the ip address and port of the TCP server, taken from groupbox **“WiFi Settings”** in PC utility window program.

If everything goes well, you will get a “connected!” message, and then you can directly run and use any utility you want. If there is a problem connecting to the given Access Point or TCP server, you will be prompted for retrying or give new settings.

If for any reason you want to change TCP server’s port number, you can do it from **“Wifi Settings”** group box, (you will have to close and reopen the PC utility for activating the new port setting) .

Note also that, when a WiFi Connection is established, the **Connection button** will be automatically enabled and connection type will be set to: **“Connection:WiFi”**. You can also have a serial connection and a WiFi connection both activated at the same time and select which one to use by pressing the connection type button.

In order to disable WiFi module connection you must first send a single string of: “+++” (check **AT+CIPMODE** command bellow) and then give the AT command:

**“AT+CIPCLOSE”** to disconnect from TCP server.

ESP8266 module use AT commands for control and communication.

You can give all AT commands using terminal program, or directly, using this small BASIC program (a\$ is the AT command string):

```
10 FOR k=1 TO LEN(a$)
20 OUT &FBD0,ASC(MID$(a$,K,1))
30 NEXT K
40 OUT &FBD0,13:OUT &FBD0,10
```

Note that after giving the AT command, an: **out &fbd0,13:out &fbd0,10** (CR and line feed) must be also sent for proper execution!

There is a [very big manual](#) with 100’s of commands available, but bellow, i give you the most useful commands, for configuring the module manually:

**AT+CIPMODE= 0 or 1** Sets Transmission Mode

0: normal transmission mode (must be used when WiFi acts as a TCP server)

1: UART-Wi-Fi passthrough mode (transparent transmission), this mode is used for communication with the PC utility program. It can only be enabled in TCP single connection mode Notes:

- The configuration changes will NOT be saved in flash.
- During the UART-Wi-Fi passthrough transmission, **if the TCP connection breaks, ESP8266 will keep trying to reconnect until “+++” is input to exit the transmission.** If it is a normal TCP transmission and the TCP connection breaks, ESP8266 will give a prompt and will not attempt to reconnect.

**AT+CIPSEND** Start sending data in transparent transmission mode.

Data must be sent with a **20-ms interval between each packet**, and a **maximum of 2048 bytes per packet**.

To **disable** transparent transmission mode, a **single packet** containing ‘+++’ must be received, (in practice, you must give: **out &fbd0,43:out &fbd0,43:out &fbd0,43**) ESP8266 returns to normal AT command mode. Wait for at least one second before sending the next AT command.

This command can only be used in transparent transmission mode which requires single

connection. Note also that, **you CANT give AT commands when transparent transmission mode is activated.**

**AT+CIPSEND=[link id],x** send x bytes. Bytes/chars follow immediately after the command. Link id is usually 0 for single connections, but can be up to 5 in a LAN situation

**AT+CIPCLOSE** Closes the TCP/UDP/SSL Connection.

**AT+CWLAP** Will list access points in range.

**AT+CWJAP="yourSSID","yourWifiPassword"** connects to an access point

**AT+CWQAP** quit from access 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** (1 for standard mode , 2 for AccessPoint mode & 3 for both)

**AT+CIPMUX=1** (set multiple connections, this must be set to '1' if you want to use module as TCP server)

**AT+CIPSERVER=1,80** (start the TCP server at port 80)

**AT+CIFSR** (shows WiFi module's ip/mac address)

**ATE0** remove echo (recommended for having WiFi module act as TCP server)

**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**

Example: **AT+CWSAP\_DEF="ESP8266","1234567890",1,3**

For more detailed information and examples, you can refer to the AT command manual mentioned above.

## 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 disk image disk (you can transfer it to a real disk using [CPCDISKXP](#)), and of course the PC utility program.