Users Manual

HDA800 Evaluation Kit

SPB800 Serial to WiFi solution

H&D Wireless

Revision History

Revision	Revision date	Description
PA1	2010-11-30	First Issue
PA4	2010-12-17	Updated for revision R2B of PC Connection Board
PA5	2011-01-12	Added XPLAIN example

CONTENTS

1 Evaluation Kit Content

- SPB800E, SPB800 Extension board with a 10pos. header socket, to allow easy plug in.
- PC connection board
- USB cable
- USB Flash memory with documentation and reference code

2 Preparations

To communicate with the SPB800E from a PC you need the PC connection board that convert the SPB800E's UART signal to USB or RS-232 and provide 3.3V power to the module.

On the PC you need a serial communications application, such as TeraTerm or HyperTerm. TeraTerm can be downloaded from http://www.ayera.com/teraterm/download.cfm

2.1 USB driver installation

If you are running Windows 7, it usually recognize the serial to USB chip and installs the drivers at first connection. For other operating system there are driver software provided on the USB memory under the directory called "USB_driver"

If you cannot find the appropriate driver for your system or having other problems with the USB driver please seek more information at http://www.ftdichip.com/Drivers/VCP.htm

Remember to set the two jumpers on the PC connection in their right position to enable USB as serial port. If you rather use the RS-232 port the jumpers should be in the left position. See Figure 2-1.

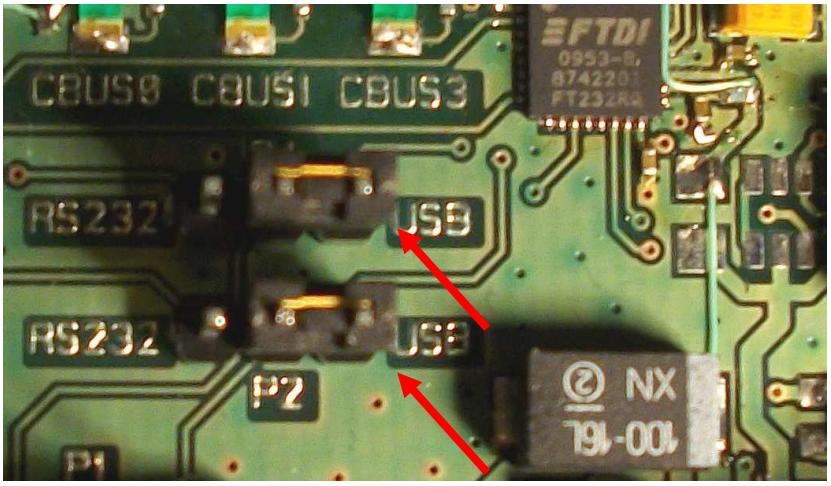


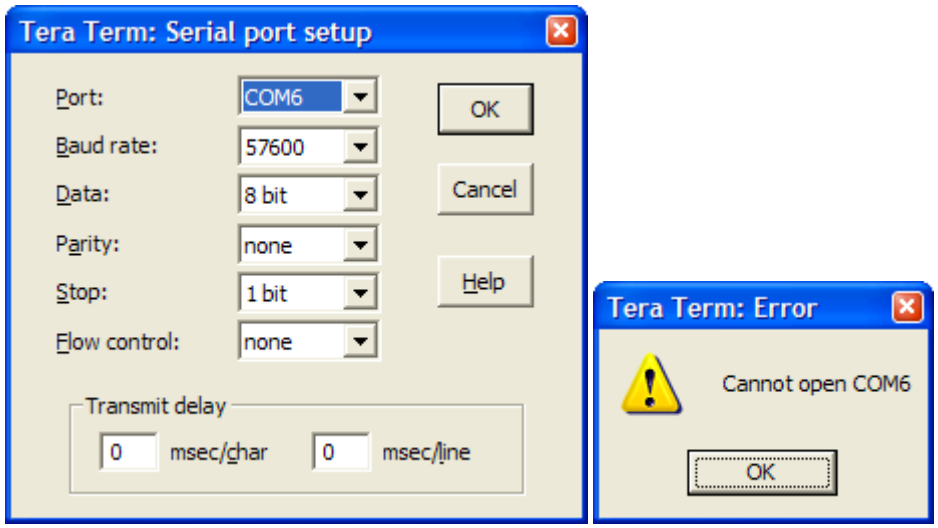
Figure 2-1: Port selection jumpers

3 Connecting the Kit

Connect the Evaluation Kit as follows:

1. Plug in the SPB800E into the 10-pin header.
2. Connect the PC Connection board with the USB cable to a USB port on a PC. The USB powers the board so no other power source is required.
   Or alternatively:
   Power the PC Connection board from an AC/DC adapter with between 5V – 9V DC output and plus on the sleeve and negative on the centre pin. Connect with a serial cable to the PC.

3. Start a serial communications application such as TeraTerm on PC1.
   In TeraTerm, select "Setup->Serial Port…" in the menu and configure the serial port according to the figure below.



   Note that the port number used to communicate with the SPB800E (COM6 in the figure above) might be different in some environments, so make sure to try another one if COM6 fails (see figure below)
   The assigned port can also be found in the Windows Device Manager, under Ports (COM & LPT).

4. Press the reset button on the PC Connection Card to restart the SPB800E

5. In the TeraTerm window a series of dots appears one every second

6. Hit "Return" in the TeraTerm terminal within 5 seconds from the first dot to keep the SPB800E in configuration mode. A $ prompt should show in the terminal.

   Now you are ready to configure the SPB800E for one of the examples

4 SPB800 Modes

The SPB800 are designed to operate in two different modes, one where the host is aware of the SPB800 and uses the oWL-pico API to configure and control the SPB800, another where the host is unaware of the SPB800 which is pre-configured and just forwards data on the serial port onto a TCP socket.

The two modes are controlled by the parameter /proto/active. For oWL-pico mode the parameter is set to *active* and for the un-aware mode to *none.*

5 Commands

The following commands are available for the SPB800 in configuration mode. For (o)WL-pico API command please see the (o)WL-pico API Specification.

db	Read/write database
upgrade	fw upgrade xmodem
reset	reset device
help	print a list of available commands

The db (data base) command splits in several commands

db reset [path]	reset param(s) to default (need to be followed by a db store)
db get [path]	list parameter(s) in the edit list.
db set <path> [args ...]	set single parameter
db load	read parameters from non-volatile memory (flash)
db store	write parameters to non-volatile memory (flash)

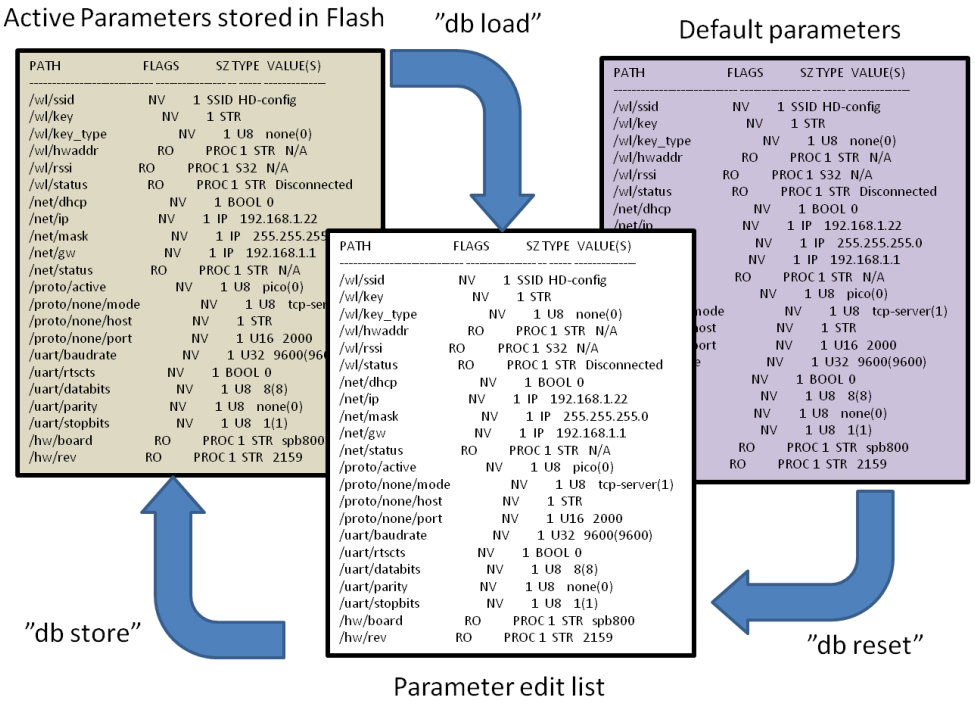


Figure 5-1: Parameter data base

At start up the SPB800 read the parameters from the non-volatile memory, see Figure 5-1. To change a parameter value it needs to be changed in the edit list and then stored in non-volatile memory with the db store command. “db store” stores all the values in the edit list at the same time. The current parameters in non-volatile memory can be retrieved with the command “db load”. Factory default setting can be retrieved with the command “db reset”. Please not that to restore the active settings to factory default the parameters also have to be stored with “db store”.

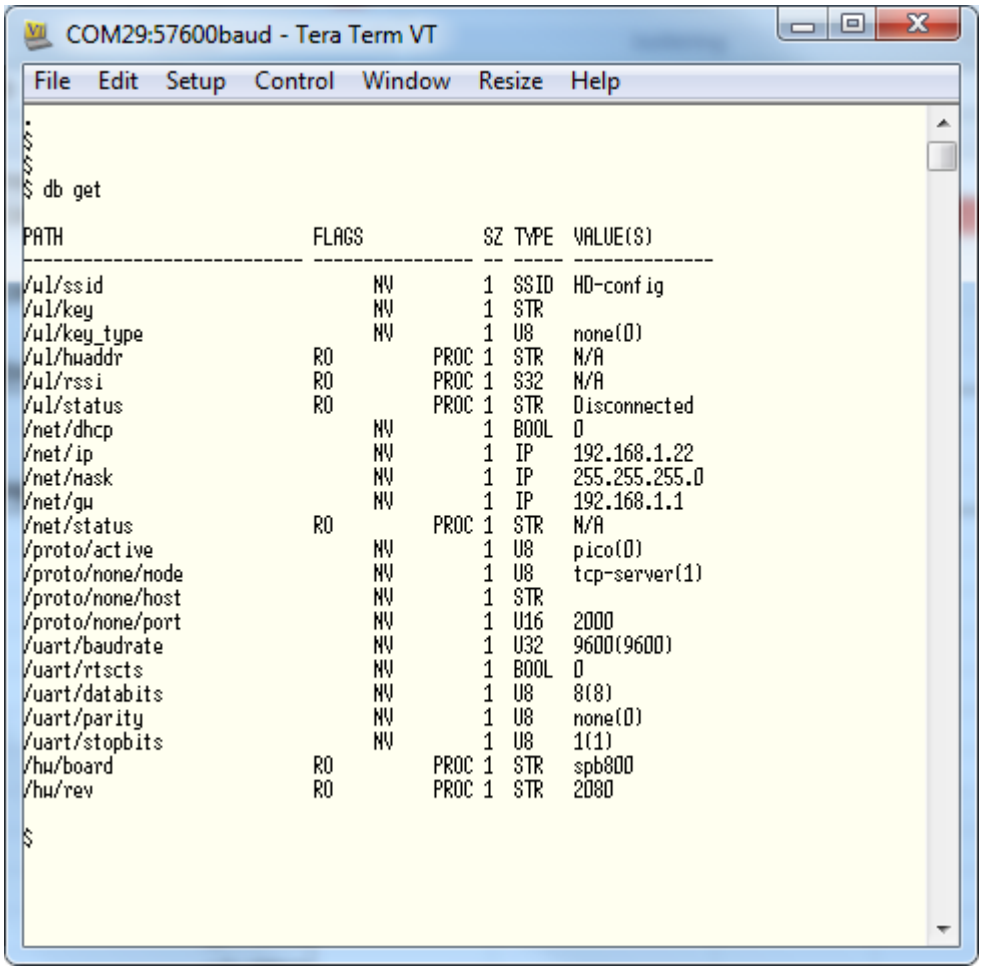


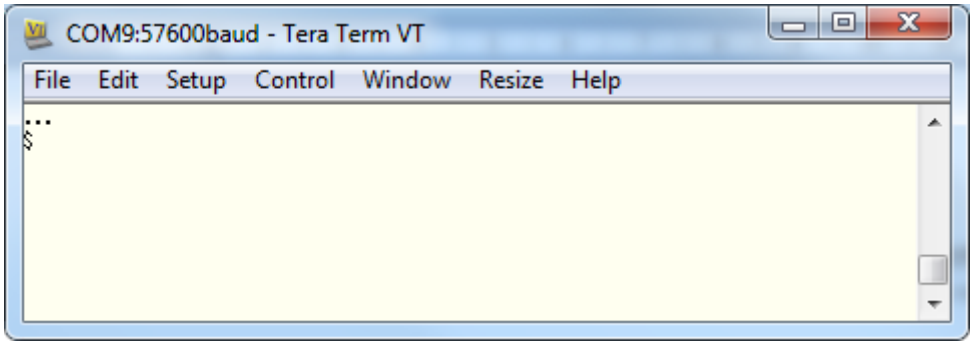
Figure 5-2: Sample of db get listing

6 Example 1, Serial port Wireless LAN adapter

This example describes how to configure the HDA800 Evaluation Kit to a RS-232 – Wireless LAN adapter. It makes use of a direct method changing the parameter database values with commands from a consol terminal like TeraTerm or similar. This exemplify how the SPB800 can be used with a un-aware host with the (o)WL-pico API not active, i.e. the parameter "/proto/active" set to "none".

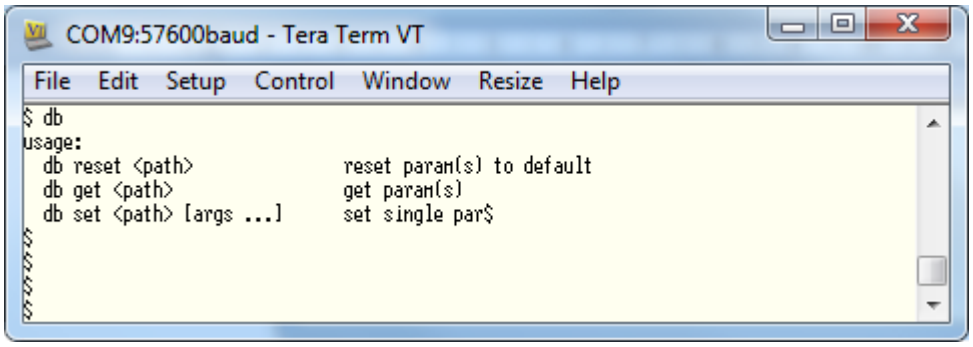
How to do it:

- Start a serial port terminal on the PC
- Connect the HDA800 to the Serial Port/Serial port adapter on the PC
- Insert the DC plug to power the unit and hit "Return" within 5 seconds. You should read a string of dots .. on the terminal and a $ prompt once you have hit return. If you are to late in hitting "Return" the terminal won't do anything.



Example of terminal printout

- The following commands are available



- With "db set" you can set all parameters for the HDA800 to connect to a WiFi Network as well as configure the serial port settings.

Parameter	Values	Command
IP Network parameters		
IP address	IPv4 address	db set /net/ip <ip>
Default Gateway	IPv4 address	db set /net/gw <gw>
Netmask	IPv4 address	db set /net/mask <mask>
Enable DHCP	Boolean	db set /net/dhcp <0 or 1>
Wireless LAN parameters		
SSID	String	db set /wl/ssid <ssid>
Type of key used	none, wep, wpa	db set /wl/key_type <none, wep, or wpa>
WEP, WPA/WPA2 key	String	db set /wl/key <key>

RS-232 parameters		
Baudrate	9600, 38400, 57600, 115200, 230400	db set /uart/baudrate <rate>
Data	5, 6, 7 or 8 bit	db set /uart/databits <5,6,7 or 8>
Parity	even, odd, none	db set /uart/parity <even, odd or none>
Stop bits	1, 2	db set /uart/stopbits <1 or 2>
Flow Control	Boolean	db set /uart/rtscts <0 or 1>
Application parameters		
Activate oWL-pico	none, pico	db set /proto/active none (should always be set to none for RS232 adapter mode)
TCP socket	String	db set /proto/none/port
Server/Client	tcp-server, tcp-client	db set /proto/none/mode
Server IP address	IPv4 address	db set /proto/none/host <ip>

- Typing an erroneous command will list the available command as a help

Configuration Example:

- In this example we want to configure the HDA800 for a WiFi Network with the following characteristics
  - SSID = my-wifi-net
  - Encryption = WPA
  - Encryption passphrase =my-secret-key
  - IP address we want to assign = 192.168.2.10
  - Subnet mask = 255.255.255.0
  - Default Gateway 192.168.2.254

- Important! No settings are stored in the flash memory until the command “db store” is given.

- First we need to set the parameter /proto/active to none

  $ db set /proto/active none

- Then we continue with entering the SSID

  $ db set /wl/ssid my-wifi-net

- Then we enter the security encryption type in this case we are using WPA

  $ db set /wl/key_type wpa

- Enter the key

  $ db set /wl/key my-secret-key

- Then we disable the DHCP client as we want to set a static IP address for the HDA800.

  $ db set /net/dhcp 0

- We enter the IP address we have selected for the HDA800. Make sure that is does not collide with any other static set address or the address range managed by the networks DHCP server.

  $ db set /net/ip 192.168.2.10

- We enter the corresponding network mask.

  $ db set /net/mask 255.255.255.0

- Then we enter the default gateway of the network

  $ db set /net/gw 192.168.2.254

- We then in a similar fashion set the RS-232 parameters, first the baud rate to 9600 kbps

  $ db set /uart/baudrate 9600

- We turn flow control off

  $ db set /uart/rtscts 0

- The number of data bits to 8

  $ db set /uart/databits 8

- The number of parity bits to none

  $ db set /uart/parity none

- The number of stop bits to 1

  $ db set /uart/stopbits 1

- Finally we defines the HDA800 as server for the TCP socket

  $ db set /proto/none/mode tcp-server

- And set the port of the TCP socket to 2001

  $ db set /proto/none/port 2001

- To check our parameters we type “db get” to list all parameters to check.

  $ db get

PATH	FLAGS	SZ	TYPE	VALUE(S)
/wl/ssid	NV	1	SSID	my-wifi-net
/wl/key	NV	1	STR	my-secret-key
/wl/key_type	NV	1	U8	wpa(2)
/wl/hwaddr	RO PROC	1	STR	N/A
/wl/rssi	RO PROC	1	S32	N/A
/wl/status	RO PROC	1	STR	Disconnected
/net/dhcp	NV	1	BOOL	0
/net/ip	NV	1	IP	192.168.2.10
/net/mask	NV	1	IP	255.255.255.0
/net/gw	NV	1	IP	192.168.2.254
/net/status	RO PROC	1	STR	N/A

/proto/active	NV	1	U8	none(0)
/proto/none/mode	NV	1	U8	tcp-server1)
/proto/none/host	NV	1	STR	
/proto/none/port	NV	1	U16	2001
/proto/pico/ascii	NV	1	BOOL	0
/proto/pico/echo	NV	1	BOOL	0
/uart/baudrate	NV	1	U32	9600(9600)
/uart/rtscts	NV	1	BOOL	0
/uart/databits	NV	1	U8	8(8)
/uart/parity	NV	1	U8	none(0)
/uart/stopbits	NV	1	U8	1(1)
/hw/board	RO	PROC 1	STR	spb800
/hw/rev	RO	PROC 1	STR	N/A

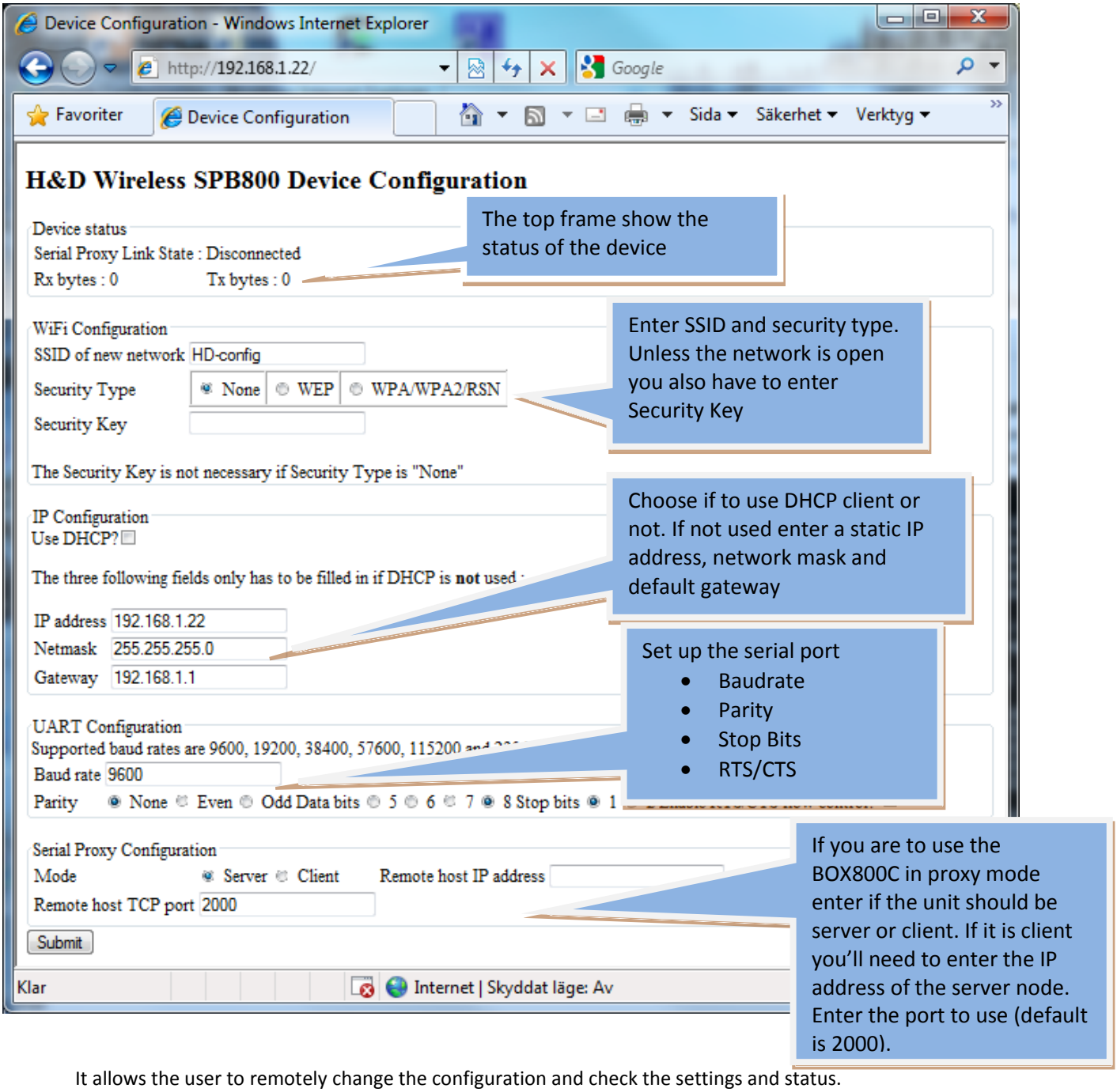
- Satisfied with the result we enter the parameters into the flash with ”db store”

  $ db store

- We can now disconnect the HDA800 from the serial port and DC power and connect it to any equipment that communicates with a RS-232 serial port and access that through a TCP socket on port 2001.
- See also *1543- HDA800 Quick Start Guide* for more information on this example.

6.1 Web Configuration

When the pico mode is turned off (/proto/active = none) the SPB800 publish a configuration web page at its IP address. As default the SPB800 is configured for an open netweork with SSID = HD-config and IP address 192.168.2.22.



It allows the user to remotely change the configuration and check the settings and status.

7 Example 2, Linux emulation of host

In pico mode, the SPB800 device is controlled by a host through a serial connection. The host is typically an embedded platform where wifi and ip functionality should be added through the SPB800 and the oWL-Pico API.

Included in the release is a reference design application which uses the oWL-Pico API. The reference design provides an interactive shell where the various functions of oWL Pico API can be invoked.

The reference design supports linux hosts and the ATMEL Xplain board.
Both source and binaries are provided for those platforms. See the section about compiling for information on how to recompile the reference design. To get started quickly, the binaries can be used.

7.1 Running the reference design on linux

- First make sure that the pclient binary is executable by issuing the following command:

  $ chmod +x pclient

- Then power up the SPB800 board and connect the eval kit to the PC. Allow the device to boot up completely before starting the pclient application on the linux host.

- On a linux host then start the pclient application with the following command:

  $ ./pclient /dev/ttyUSB1 115200

- /dev/ttyUSB1 should be replaced by the actual serial port on the linux host where the SPB800 EVK is connected.

- The help command will list the available commands:

```
$ help

send_file          send_file <sockid> <file>
recv_file          recv_file <sockid> <file>
recv               recv <sockid>
linkup             wlp_linkup(ssid, key, wep)
linkdown           wlp_linkdown()
get_hwaddr         wlp_get_hwaddr()
get_network        wlp_get_network()
set_ipaddr         wlp_set_ipaddr(ip, mask, gw, dns)
get_ipaddr         wlp_get_ipaddr()
set_dhcp           wlp_set_dhcp(enable)
get_dhcp           wlp_get_dhcp()
get_hostbyname     wlp_get_hostbyname(dnsname)
socket             wlp_socket(type, proto)
bind               wlp_bind(sock, ip, port)
listen             wlp_listen(sockid, backlog)
connect            wlp_connect(sockid, ip, port)
close              wlp_close(sockid)
send               wlp_send(sockid, data)
sendto             wlp_sendto(sockid, data, ip, port)
get_peeraddr       wlp_get_peeraddr(sockid) ttcp
```

```
ttcp              throughput test
help              print this information
```

- See the command reference for information about each command and the examples section for information about how to combine commands.

7.2 Testing the HTTP server implemented in the pclient reference design

- Use the command "linkup" to connect to an access point on the local network. The ssid, encryption key and optionally encryption type should be specified:

```
$ linkup my-ssid my-key123 0
conn_cb: 1
```

- In the case above, the ssid of the access point is "my-ssid", the key is "my-key123" and the key type is wpa (0). If a wep key is used, the key type should be 1.
- The "conn_cb: 1" will be printed as soon as the link is up.
- If the access point supports dhcp, an ip address can be requested with the set_dhcp command:

```
$ set_dhcp 1
addr_cb: 192.168.2.111
```

- The "addr_cb: 192.168.2.111" will be printed when an ip address has been obtained.
- If the access point don't support dhcp, a static ip address configurion can be set with the set_ipaddr command:

```
$ set_ipaddr 192.168.2.10 255.255.255.0 192.168.2.1 192.168.2.1
```

- In this case the ip address was set to 192.168.2.10, the network mask to 255.255.255.0, the gateway to 192.168.2.1 and the dns server to 192.168.2.1.
- When an ip address has been configured, the pclient software will start an http server. This server can be accessed by browsing to the configured ip address using a web browser on a PC connected to the same network.

7.3 Compiling the reference design

- The reference design software can be compiled for a linux host or for the ATMEL Xplain board.
- To compile for linux, run the following commands:

```
$ cd owl_pico-r2033/ports/linux
$ make
```

- The binary "pclient" should now be created.
- To compile for Xplain, asf-2.0.0 from atmel must first be extracted. Then run the following commands:

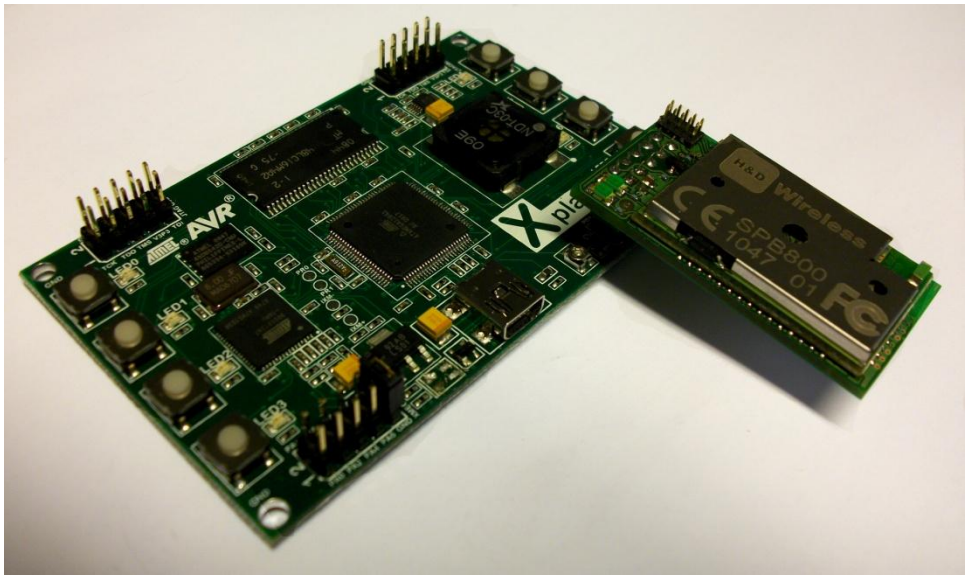
```
$ cd owl_pico-r2033/ports/linux
$ make AVR8_FRAMEWOR=/path/to/asf-2.0.0/xmega
```

- Note that /path/to should be replaced by the actual location where asf-2.0.0 was extracted.
- The binary "pclient.hex" will be produced which can now be flashed onto the Xplain board. Exactly how to perform this is out of the scope of this manual.

8 Example 3, (o)WL-Pico on Atmel XPLAIN

In this example a reference design with the (o)WL-pico client is run on Atmel's evaluation and demonstration kit for ATxmega128A1, Xplain.

- Compile the reference design for Xplain, see 7.3, or download the .hex file from www.hd-wireless.se
- Load the reference design binary file, pclient.hex, onto the XPLAIN board
- The SPB800E should be connected to the lower right pin header on the XPLAIN board



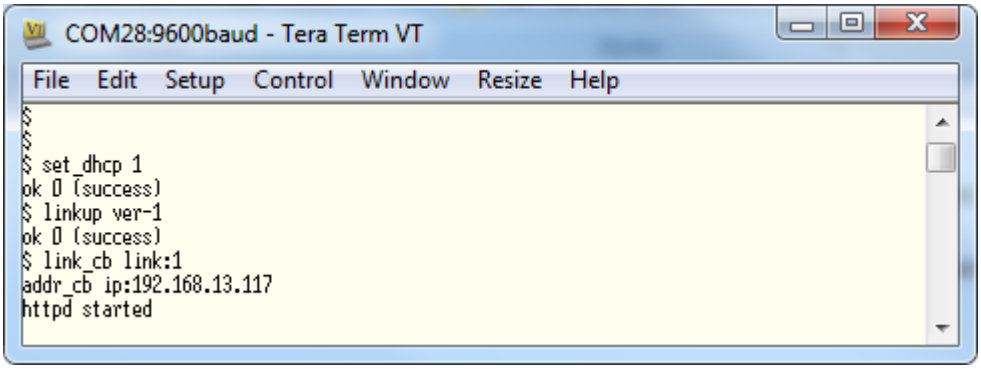
- Connect the XPLAIN board to a USB port on a PC.
- Open a serial port terminal (9600, 8, no, 1, none) and type the oWL-pico API command

  $ set_dhcp 1

- This will enable the DHCP client in the SPB800
- The give the command

  $ linkup my-ssid (my-key 0)

- Where "my-ssid" is the ssid you want to connect to. It is optional to add key and security type (0 = WPA, 1 =WEP), to connect to a secure network.
- The SPB800 will connect to the WLAN with the set ssid and light one LED when successful and confirm the connection back to the host (XPLAIN)
- Then its DHCP client requests an IP address and when it is received it lights the second LED and reports the acquired IP address to the host.



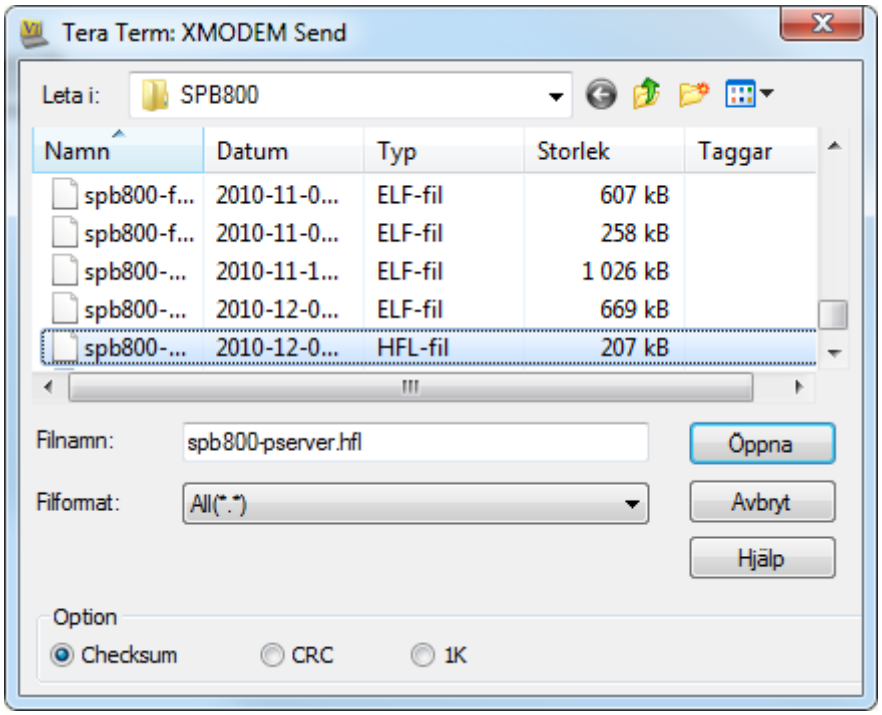
- Now you can open the webpage with a web-browser on any platform (PC, iPad, SmartPhone) that is connected to the same network

- The webpage allows wireless control over the LEDs on the XPLAIN and takes a reading of the built in temperature sensor.

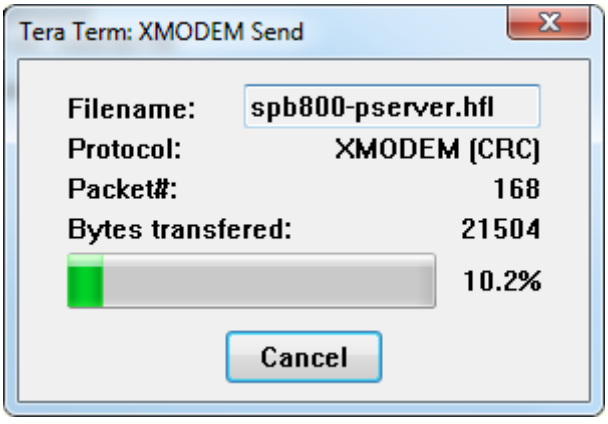
9 Firmware Upgrade

To upgrade the firmware of the SPB800E do as follows:

- Connect the SPB800E with the PC Connection board to a Serial port of a PC and start TeraTerm (57600 8N1)
- Power up or power cycle the SPB800E and hit return in TeraTerm within 5 seconds from the first dot is visible in the terminal window.
- Enter configuration mode by pressing <enter>
- Type "upgrade". SPB800 will now wait for a file being transferred using the XMODEM protocol.
- In TeraTerm, select File->Transfer->XMODEM ->send and select the firmware image (SPB800-pserver.hlf). Make sure the "checksum" radio button is selected.



- The transfer should start when OK is pressed. Note that SPB800 will time out after about 20 seconds if not file is sent, in that case restart from typing upgrade and try again.



- When the transfer is complete, the SPB800 will print "completed - rebooting" in the TeraTerm window. Now wait until the dots are printed again.