IP2022 Internet Processor™

Advanced Wireless Kit

User's Guide

Revision History

Revision	Release Date	Summary of Changes
1.0	September 16, 2002	First issue

Part #:: IP2K-DUG-ADVWLESS-10

© 2002 Ubicom, Inc. All rights reserved. No warranty is provided and no liability is assumed by Ubicom with respect to the accuracy of this documentation or the merchantability or fitness of the product for a particular application. No license of any kind is conveyed by Ubicom with respect to its intellectual property or that of others. All information in this document is subject to change without notice.

Ubicom products are not authorized for use in life support systems or under conditions where failure of the product would endanger the life or safety of the user, except when prior written approval is obtained from Ubicom.

Ubicom™ and the Ubicom logo are trademarks of Ubicom, Inc.
Internet Processor™ is a trademark of Ubicom, Inc.
All other trademarks mentioned in this document are property of their respective companies.

Ubicom, Inc.
635 Clyde street
Mountain View, CA 94043
tel 650 210 1500
fax 650 210 8715
www.ubicom.com

Table of Contents

List of Figures

Preface

This manual introduces the Advanced Wireless Kit — hardware and software tools used for wireless applications development on the IP2022 Internet Processor. The Advance Wireless Kit includes the Universal Device Networking Kit and supplements it with additional hardware and software components. For basic information about using the Universal Device Networking Kit, refer to the *Universal Device Networking Kit User's Guide.* For detailed information about programming the IP2022, see the *IP2022 Programmer's Reference Manual* .

Related Documentation

Main documentation for the IP2022, available from Ubicom:

- *IP2022 Data Sheet*, available from Ubicom.
- *IP2022 Programmers Reference Manual*.
- IP2022 *Universal Device Networking Kit User's Guide.*

Notational Conventions

In this document, the notation "->" is used to refer to a command selected from a menu. For example, the Save command on the File menu is File -> Save.

The Programs menu is accessed by clicking on the Start button (lower left corner of screen), then clicking on Programs. After installing the Ubicom software, the Programs menu will contain a Ubicom entry which is used to access the software tools. In this document, references to the Ubicom menu actually mean commands selected from the Start -> Programs -> Ubicom menu.

File Naming Conventions

Both MS-DOS and Unix file naming conventions are used in this document. An MS-DOS file name uses backslashes as separators, such as C:\Ubicom\sdk\projects\starter\Makefile.

Unix file names are used for Configuration Tool parameter values, names in **`make`** files, and the SDK directory tree. A Unix file name uses forward slashes as separators, such as /cygdrive/c/Ubicom/sdk/projects/starter/Makefile.

The Unix operating system is case sensitive, e.g. the names “makefile” and “Makefile” would refer to two different files. Because the software tools run under Windows/MS-DOS, however, all file names are interpreted as non-case-sensitive without regard to which file naming convention is used.

The tools and utilities of this kit do not support file names that have embedded space characters; so, names like “Program Files” cannot be used as names for files or directories in the path to a file.

Chapter Summary

Chapter 1 is a quick procedure to set up the Advanced Wireless Kit and verify that it is operating correctly.

Chapter 2 describes the components of the kit

Chapter 3 introduces the example wireless projects that demonstrate use of the IP2022 in wireless applications.

Chapter 4 presents the UI Generator program and Elf Editor program.

Chapter 5 identifies where to make the most common changes the parameters of the example projects.

Appendix A contains schematic diagrams of the various units of the Demo Board.

Glossary

802.11b — An IEEE standard for 11Mbps wireless Ethernet

Access Point (AP) — a hardware device or computer software that acts as a communication hub for connecting users of a wireless device to a wired LAN. APs are important for providing heightened wireless security and for extending the physical range of service a wireless user has access to.

Bridge — connects a wireless network to a wired network transparently. Communication is possible between both networks in both directions.

CompactFlash — a stardard for matchbook sized removable memory and I/O devices. See also www.compactflash.org.

PC Card — a stardard for credit card sized removable memory and I/O devices developed by PCMCIA.

PCMCIA — *Personal Computer Memory Card International Association* which has developed the PC Card standard for small peripheral devices. See also www.pcmcia.org.

Station — a wireless client or end point.

Wi-Fi — a registered trademark (short form of “wireless fidelity”) of the Wireless Ethernet Compatibility Alliance (www.weca.net) for designating devices that are certified compatible with IEEE standard 802.11b.

10 Quick Set Up

The following steps comprise a quick set up procedure for the CD-ROM software and PCMCIA daughter board.

1. *Register Your Kits* — at Ubicom's Technical Support Portal (www.ubicom.com). Be sure to register both the IP2022 Universal Device Networking Kit and the IP2022 Advanced Wireless Kit upgrade. This provides access to latest documentation and software available for these kits. Look in the Downloads section for software releases more recent than those of your CD-ROMs.
2. *Install the Universal Device Networking Kit* first — Follow the "Quick Set Up" procedure from the *IP2022 Universal Device Networking Kit User's Guide* to install both the software and IP2022 Demo Board.
3. *Install Advanced Wireless Kit Software* — Run the installation program from the CD ROM (or from the support portal, whichever is more recent):

   `\Install\Ubicom_ADVWLESS_4.2.exe` (the name may vary)

4. *Configure the Demo Board* — With power off, install jumper JP3 to enable flash memory. Also make sure that JP7 is installed to enable the on-board LEDs, P10 set to 3.3V, JP13 to lower pins (oscillator), JP14, JP15, JP16, JP17 (power supplies). Make the following connections:
   - Insert the 802.11 PC Card into the PCMCIA daughter card and plug the daughter card into connector J3 of the Demo Board.
   - Insert the ethernet daughter card in connector J2.
   - Connect the ethernet daughter card to a hub or PC (no IP configuration; only an Ethernet link is required)
   - Connect the programming dongle as described in the *Universal Device Networking Kit User’s Guide*.

   Power on the board.
5. *Create a Self-Test Project* — create a directory called **`C:\SDK_Demo\self_test`**. Open Unity by selecting Programs -> Ubicom -> Unity from the Windows Start menu. Then, select Project -> New from the Unity menu bar and navigate to the new directory. For the project name, enter **`self_test.c_c`**. For the project type, select SDK, and click the OK button. Select template **`self_test`**, then click the OK button.
6. *Configure the Project* — (optional; only to change the default configuration) From the Unity menu bar, select Tools->Configure, which loads the file **`self_test\config\self_test.lpj`** into the Configuration Tool.
   - Generate the header files that pass configuration parameters to the build process (menu Package -> Generate).

- Close the configuration tool.

7. *Compile Project* — Select Build -> Compile from the Unity menu bar.
8. *Download Project* — Enter device programming mode by selecting the Build -> Start Programmer command from the Unity menu bar. In the new window which appears, click the Program button. When the “Complete OK” message appears, click the Close button.
9. *Verify Operation* — This program tests external flash, Ethernet (connected to a hub/switch), WLAN PC-Card, and indirectly the IP2022. It lights LEDs to communicate status. During the test, LEDs D7, D8 and D9 should flash. After the test, these same 3 LEDs should be on; if not, something is not right, and the LEDs indicate the problem area:
   - LED D7 OFF: Ethernet link failed
   - LED D8 OFF: External serial flash failed
   - LED D9 OFF: PCMCIA interface failed

2.0 Overview

2.1 Minimum System Requirements

The basic requirements are the same as those of the Universal Device Networking Kit as defined in the *Universal Device Networking Kit User's Guide.* In addition, some of the demonstation projects require complementary wireless devices with which to communicate:

- wlan_bridge project — an access point is needed to exercise the bridge.
- wlan_ap project — requires stations (either PC Cards in note-books computers or another Ubicom wlan_bridge)
- wlan_rftest project — another wlan_rftest platform is needed, so one can transmit while the other is receiving.

Ubicom offers 802.11b WLAN evaluation kits that can also be configured and customized using this development kit. The boards are standalone, requiring no daughter cards and contain only the components necessary for Ethernet to 802.11b WLAN support.

- 802.11b PCMCIA Eval Kit (part #IP2K-KEV-11BPCM) contains:
  802.11b PCMCIA Eval Board (part #IP2K-BEV-11BPCM)
  802.11b WLAN PC-Card (part #UBI-FGC-11BPCM)
  5V DC Power Supply

- 802.11b CF Eval Kit (part #IP2K-KEV-11BCF) contains:
  802.11b CF Eval Board (part #IP2K-BEV-11BCF)
  802.11b WLAN CF Card (part #UBI-FGC-11BCF)
  5V DC Power Supply

2.2 Installing the Software

The installation file **`Ubicom_ADVWLESS_4.2.exe`** (name may change with future releases) is an upgrade to the Universal Device Networking Kit. It adds a mixture of source code, object code, sample projects, and addtional development tools — all designed to help with wireless applications develoment.

Install the Advanced Wireless Kit software after installing the Universal Device Networking Kit software according to the instructions in the *Universal Device Networking Kit User’s Guide.*

2.3 New Contents of Ubicom Directory

The following software components are added by the Advanced Wireless Kit installation:

- ipOS packages for wireless applications
  - sdk/pkg/ipWLANstation
  - sdk/pkg/ipWLANaccesspoint
  - sdk/pkg/ipBridge

  Refer to the SDK Help for documentation on these packages.
- Demonstration applications
  - sdk/projects/wlan_bridge (Bridge Project)
  - sdk/projects/wlan_ap (Access Point Project)

  - sdk/projects/wlan_rftest (RF Testing Project)
  - sdk/projects/selftest (Self Test Project)
- Tools and Utilities
  - sdk/tools/uigen.exe (UI Generator)
  - sdk/tools/elfedit.exe (Elf Editor)
  - sdk/tools/elfaddfile.exe (Elf Editor helper)

3.0 Example Wireless Projects

The Advanced Wireless Kit adds several wireless LAN demonstration projects to the directory **`\Ubicom\sdk\projects`**. This chapter describes how to use the 802.11b PCMCIA daughter card together with the Demo Board and SDK of the Universal Device Networking Kit to prepare and run the following projects:

- Access Point Project
- Bridge Project
- Radio Frequency Testing Project
- Self Test Project

3.1 Requirements

The example projects require the following hardware and software:

- Personal Computer with MS Windows and an Ethernet NIC.
- TFTP client (many Windows systems have TFTP already installed).
- Universal Device Networking Kit software installed on PC.
- Advanced Wireless Kit software installed on PC.

- IP2022 Demo Board with power supply and with programming dongle and cable istalled according to instructions in *Universal Device Networking Kit User’s Guide*.
- Ethernet daughter card with integrated magnetics.
- Ethernet CAT5 cross-over cable and/or straight-through cable.
- PCMCIA WLAN daughter card.
- 802.11b PC Card.

3.1.1 Demo Board Configuration

Configure the Demo Board as follows: With power off, install jumper JP3 to enable flash memory. Also make sure that JP7 is installed to enable the on-board LEDs, P10 set to 3.3V, JP13 to lower pins (oscillator), JP14, JP15, JP16, JP17 (power supplies). Make the following connections:

- Insert the 802.11 PC Card into the PCMCIA daughter card and plug the daughter card into connector J3 of the Demo Board.
- Insert the ethernet daughter card in connector J2.
- Connect the ethernet daughter card either to a hub (with the straight-through CAT5 cable) or to the host PC (with the cross-over CAT5 cable) according to the requirements of each project.

Power on the board.

3.2 Bridge Project wlan_bridge

1. *Create a Bridge Project* — create a directory called **`C:\SDK_Demo\bridge`**. Open Unity by selecting Programs -> Ubicom -> Unity from the Windows Start menu. Then, select Project -> New from the Unity menu bar and navigate to the new directory. For the project name, enter **`bridge.c_c`** (do not use any other name; this name is hard coded in some of the project files). For the project type, select SDK, and click the OK button. Select template **`wlan_bridge`**, then click the OK button.
2. *Configure the Project* — (optional; only to change the default configuration) From the Unity menu bar, select Tools->Configure, which loads the file **`self_test\config\bridge.lpj`** into the Configuration Tool.
   - Generate the header files that pass configuration parameters to the build process (menu Package -> Generate).
   - Close the configuration tool.

   Factory default settings can be changed by editing **`config\wlan_bridge.lua`**.
3. *Compile Project* — Select Build -> Compile from the Unity menu bar.
4. Factory default settings can be changed with the Elf Editor.
5. *Download Project* — Enter device programming mode by selecting the Build -> Start Programmer command from the Unity menu bar. In the new window which appears, click the Program button. When the “Complete OK” message ap-

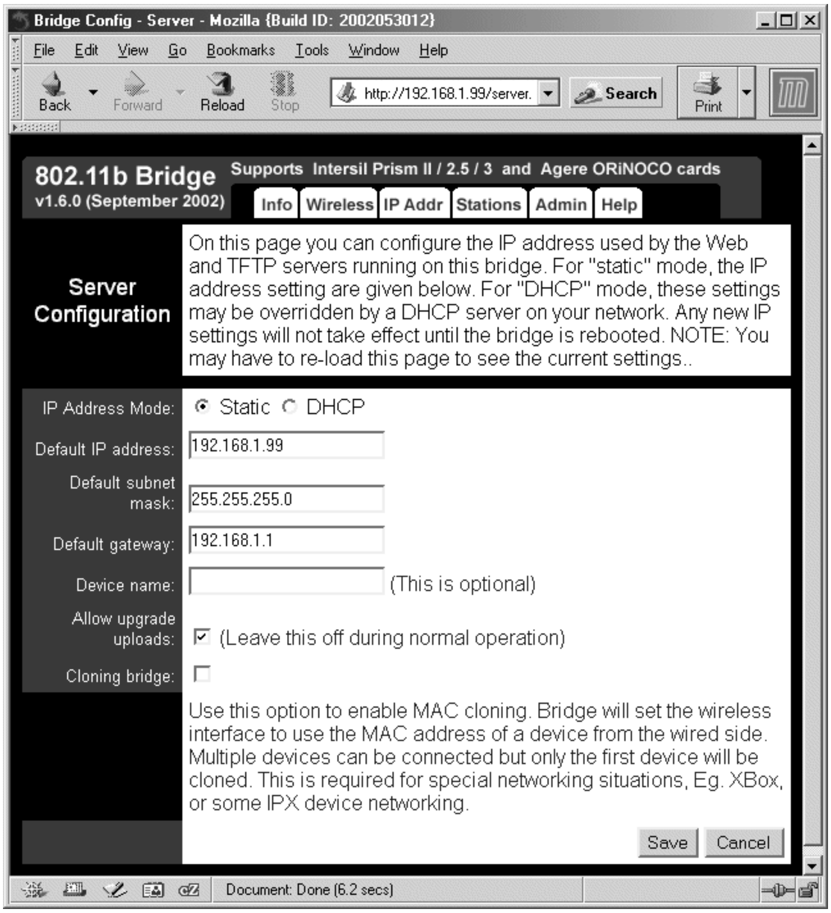
pears, click the Close button. The bridge is operational at this point, but will not be configurable without the web pages in its file system.

6. *Prepare the Web Pages File* — Open Windows Explorer and change directory to **`bridge\web_pages.`** Double click on the file **`create.bat`**. This makes a filesystem image including the web pages from the **`bridge\web_pages\files`** directory.
7. *Download Web Pages to External Flash File System* — Your PC must be configured for the 192.168.1.0/24 subnet and an Ethernet link must be extablished between the Demo Board and the PC. Open a DOS command prompt, cd to , and issue the commands:
   ```
   cd \SDK_Demo\bridge\web_pages
   tftp -i 192.168.1.99 put bridge160.bin /
   ```
8. The bridge is fully functional at this point.

3.2.1 Usage

Connect your PC to the bridge with the CAT5 Cross-over cable .

Open a browser window and direct it to http://192.168.1.99 to see the configuration Web pages, such as the one shown in Figure 3-1. By default, no username or password is required; click Enter when prompted. You may have to reconfigure your router/gateway to assign IP Addresses on this subnet. You can now configure the bridge to associate with an 802.11b Access Point.


Bridge Config - Server - Mozilla {Build ID: 2002053012}
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://192.168.1.99/server.
Search
Print
802.11b Bridge
v1.6.0 (September 2002)
Supports Intersil Prism II / 2.5 / 3 and Agere ORiNOCO cards
Info Wireless IP Addr Stations Admin Help
Server Configuration
Save Cancel
Document: Done (6.2 secs)

On this page you can configure the IP address used by the Web and TFTP servers running on this bridge. For "static" mode, the IP address setting are given below. For "DHCP" mode, these settings may be overridden by a DHCP server on your network. Any new IP settings will not take effect until the bridge is rebooted. NOTE: You may have to re-load this page to see the current settings..

IP Address Mode:	[✓ Static] DHCP
Default IP address:	192.168.1.99
Default subnet mask:	255.255.255.0
Default gateway:	192.168.1.1
Device name:	(This is optional)
Allow upgrade uploads:	✓ (Leave this off during normal operation)
Cloning bridge:	☐

Use this option to enable MAC cloning. Bridge will set the wireless interface to use the MAC address of a device from the wired side. Multiple devices can be connected but only the first device will be cloned. This is required for special networking situations, Eg. XBox, or some IPX device networking.

Figure 3-1 Bridge Web Configuration Page

3.2.2 Background

This section describes the strategy used to enable a multistation bridge (also known as “wireless workgroup bridge” and “wireless hub”) with the Intersil Prism II, Prism 2.5, or Prism 3 chipsets, which do not support this capability natively.

The key to the bridge implementation is “Layer 2.5 Address Translation”, which uses address translation within a multistation bridge to give the appearance that all clients behind the bridge use a single MAC address. The benefit of this approach is that the multiclient bridge will work with Access Points (APs) or stations from any other vendor.

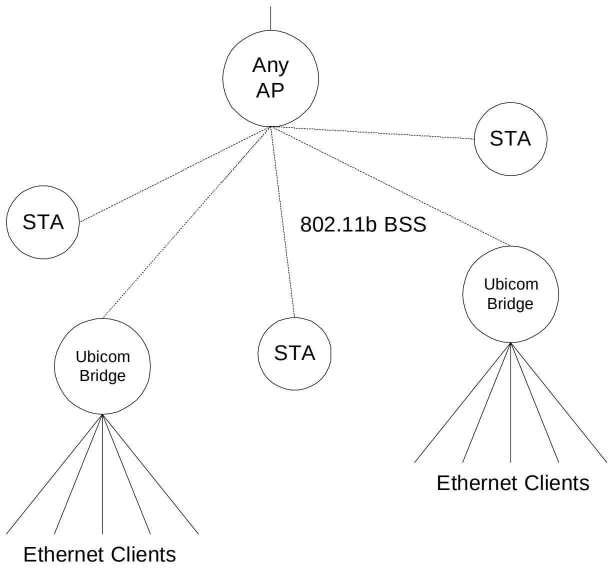


Figure 3-2 Multistation Bridge Topology

“Layer 2.5 address translation” works by associating IP addresses and MAC addresses in a table. The table is populated automatically as traffic goes from the Ethernet interface to the wireless interface. The MAC address for any packet being transmitted over the wireless interface is replaced with the MAC address of the bridge. When packets arrive on the wireless interface, the bridge looks up the IP address in the table and places the correct MAC address back into the packet header.

This capability can be used to support a number of TCP/IP based clients; however, for non-TCP/IP traffic, only one client behind the bridge can use the protocol (EtherType) at a time.

In some network configurations, using the real MAC address of an Ethernet client is required. This is the case, for example, with Microsoft X-Box and with some IPX devices. In these cases, “MAC address cloning” can be used (an option accessible from the bridge’s web page configuration). The bridge sets the wireless interface to use the MAC address of one device from the Ethernet side. Multiple devices can be connected but only the first device will be cloned.

Features

- Supports up to 255 (default configuration 50) bridged clients using TCP/IP.
- One client per non-IP protocol.
- Can be used anywhere a station can be used.
- Compatible with any 802.11b AP (infrastructure mode) or Station (ad-hoc mode).
- More than one multiclient bridge can be supported by the same AP.
- No limitations on connection initiation, devices on the Ethernet side of the multiclient bridge can act as servers.
- Implements proxy ARP.

3.3 Access Point Project wlan_ap

1. *Create an Access Point Project* — create a directory called **`C:\SDK_Demo\ap`**. Open Unity by selecting Programs -> Ubicom -> Unity from the Windows Start menu. Then, select Project -> New from the Unity menu bar and navigate to the new directory. For the project name, enter **`ap.c_c`** (do not use any other name; this name is hard coded in some of the project files). For the project type, select SDK, and click the OK button. Select template **`wlan_ap`**, then click the OK button.
2. *Configure the Project* — (optional; only to change the default configuration) From the Unity menu bar, select Tools->Configure, which loads the file **`self_test\config\ap.lpj`** into the Configuration Tool.
   - Generate the header files that pass configuration parameters to the build process (menu Package -> Generate).
   - Close the configuration tool.

   Factory default settings can be changed by editing **`config\wlan_ap.lua`**
3. *Compile Project* — Select Build -> Compile from the Unity menu bar.
4. Factory settings can be changed with the Elf Editor.
5. *Download Project* — Enter device programming mode by selecting the Build -> Start Programmer command from the Unity menu bar. In the new window which appears, click the Program button. When the "Complete OK" message appears, click the Close button.

6. *Prepare the Web Pages File* — Open Windows Explorer and navigate to the directory **`ap\web_pages.`** The files located in **`files_384x`** are dependent on the Type of MAC and the firmware stored in the WLAN card's flash memory. Ubicom provides four combinations of these files. The directory **`files_3842_121`** contains the files to be used with the 802.11b WLAN Card supplied with this kit. These files are already precopied into the **`web_pages\files`** directory.
   Double click on the file **`create.bat`**. This makes a filesystem image including the web pages from the **`web_pages\files`** directory.
7. *Download Web Pages to External Flash File System* — Your PC must be configured for the 192.168.1.0/24 subnet and an Ethernet link must be extablished between the Demo Board and the PC. Open a DOS command prompt and issue the commands:
   ```
   cd \SDK_Demo\ap\web_pages
   tftp -i 192.168.1.90 put filesystem.bin /
   ```
8. The access point is fully functional at this point.

LEDs:

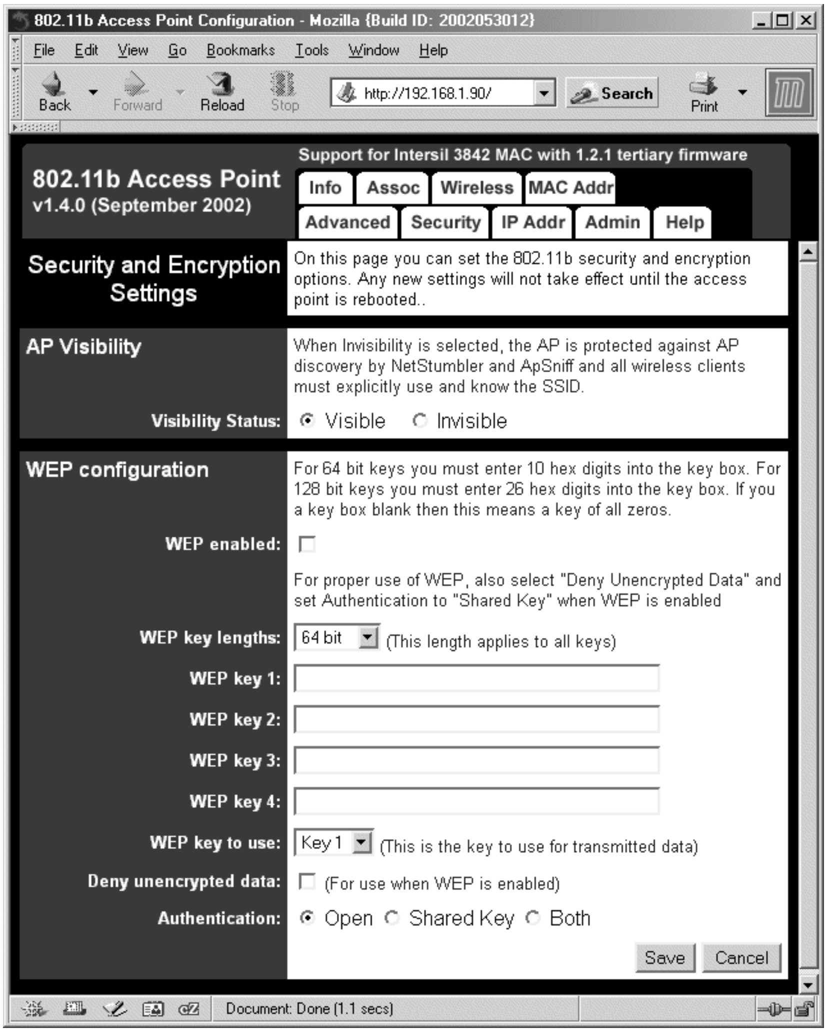
- D8 WLAN Tx & Link
- D7 WLAN Rx & Link
- D6 Ethernet Link (solid) and Activity (flashing)
- D5 Diagnostic/Error (should be off during normal operation)

3.3.1 Usage

Connect the Access Point to your LAN. Open a browser window and direct it to 192.168.1.90 to see the configuration Web pages, such as the one shown in Figure 3-3. By default, no username or password is required; click Enter when prompted. You may have to reconfigure your router/gateway to assign IP Addresses on this subnet.

Wireless stations can associate to this AP by setting their configuration to:

- SSID = either "wlandemo" or "any" (also known as the unspecified SSID) and the station will automatically find the AP.
- WEP = off


802.11b Access Point Configuration - Mozilla {Build ID: 2002053012}
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://192.168.1.90/
Search
Print
Support for Intersil 3842 MAC with 1.2.1 tertiary firmware
802.11b Access Point
v1.4.0 (September 2002)
Info Assoc Wireless MAC Addr
Advanced Security IP Addr Admin Help

Security and Encryption Settings

On this page you can set the 802.11b security and encryption options. Any new settings will not take effect until the access point is rebooted..

AP Visibility

When Invisibility is selected, the AP is protected against AP discovery by NetStumbler and ApSniff and all wireless clients must explicitly use and know the SSID.

Visibility Status: [✓ Visible] Invisible

WEP configuration

For 64 bit keys you must enter 10 hex digits into the key box. For 128 bit keys you must enter 26 hex digits into the key box. If you a key box blank then this means a key of all zeros.

WEP enabled: ☐

For proper use of WEP, also select "Deny Unencrypted Data" and set Authentication to "Shared Key" when WEP is enabled

WEP key lengths: 64 bit (This length applies to all keys)

WEP key 1:

WEP key 2:

WEP key 3:

WEP key 4:

WEP key to use: Key 1 (This is the key to use for transmitted data)

Deny unencrypted data: ☐ (For use when WEP is enabled)

Authentication: [✓ Open] Shared Key Both

Figure 3-3 AP Web Configuration Page

3.4 RF Testing Project wlan_rfttest

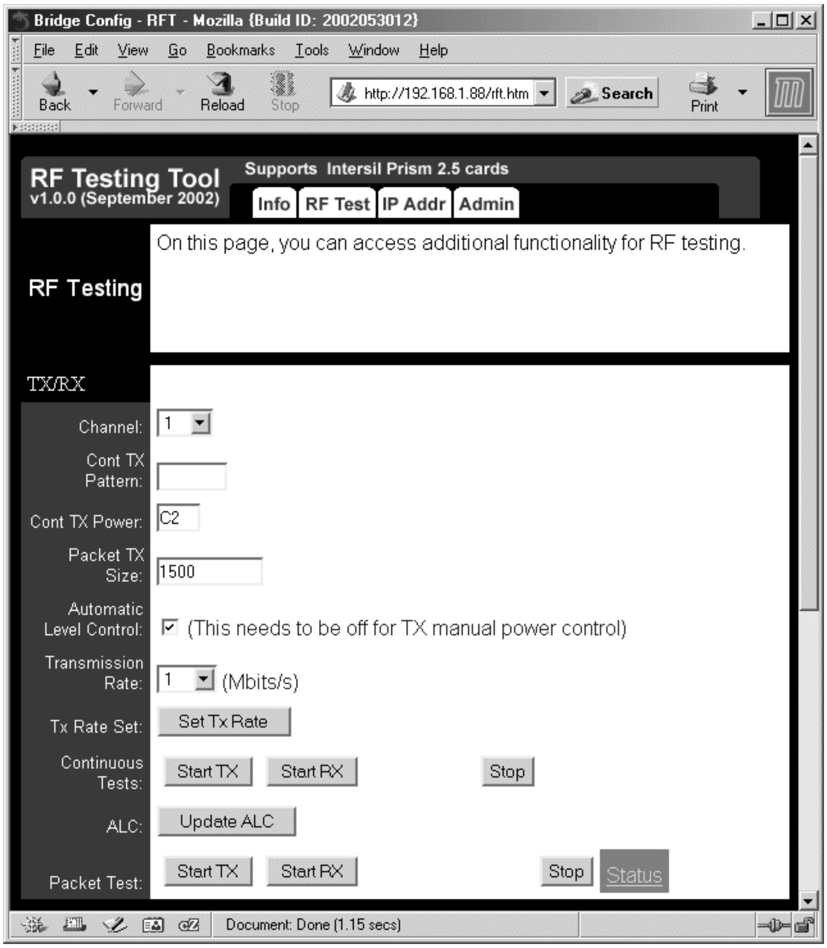
1. *Create an RF Testing Project* — create a directory called **`C:\SDK_Demo\wlan_rftest`**. Open Unity by selecting Programs -> Ubicom -> Unity from the Windows Start menu. Then, select Project -> New from the Unity menu bar and navigate to the new directory. For the project name, enter **`wlan_rftest.c_c`** (do not use any other name; this name is hard coded in some of the project files). For the project type, select SDK, and click the OK button. Select template **`wlan_rftest`**, then click the OK button.
2. *Configure the Project* — (optional; only to change the default configuration) From the Unity menu bar, select Tools->Configure, which loads the file **`config\rftest.lpj`** into the Configuration Tool.
   - Generate the header files that pass configuration parameters to the build process (menu Package -> Generate).
   - Close the configuration tool.

   Factory default settings can be changed by editing **`config\wlan_rftest.lua`**
3. *Compile Project* — Select Build -> Compile from the Unity menu bar.
4. Factory settings can be changed with the Elf Editor.
5. *Download Project* — Enter device programming mode by selecting the Build -> Start Programmer command from the Unity menu bar. In the new window which appears, click the Program button. When the "Complete OK" message ap-

pears, click the Close button. The bridge is operational at this point, but will not be configurable without the web pages in its file system.

6. *Prepare the Web Pages File* — Open Windows Explorer and navigate to the directory **`wlan_rftest\web_pages.`** Double click on the file **`create.bat`**. This makes a filesystem image including the web pages from the **`web_pages\files`** directory.
7. *Download Web Pages to External Flash File System* — Your PC must be configured for the 192.168.1.0/24 subnet and an Ethernet link must be extablished between the Demo Board and the PC. Open a DOS command prompt and issue the commands:

   ```
   cd \SDK_Demo\wlan_rftest\web_pages
   tftp -i 192.168.1.88 put rftest_web.bin /
   ```
8. You can now open a web browser at http://192.168.1.88 and access the test pages, such as the one shown in Figure 3-4.


Bridge Config - RFT - Mozilla {Build ID: 2002053012}
File Edit View Go Bookmarks Tools Window Help
Back
Forward
Reload
Stop
http://192.168.1.88/rft.htm
Search
Print
Info
RF Test
IP Addr
Admin
Set Tx Rate
Start TX
Start RX
Stop
Update ALC
Status
Document: Done (1.15 secs)

RF Testing Tool
v1.0.0 (September 2002)

Supports Intersil Prism 2.5 cards

RF Testing

On this page, you can access additional functionality for RF testing.

TX/RX

Channel:	1
Cont TX Pattern:	
Cont TX Power:	C2
Packet TX Size:	1500
Automatic Level Control:	✓ (This needs to be off for TX manual power control)
Transmission Rate:	1 (Mbits/s)
Tx Rate Set:	Set Tx Rate
Continuous Tests:	Start TX Start RX Stop
ALC:	Update ALC
Packet Test:	Start TX Start RX Stop Status

Figure 3-4 RF Testing Web Interface

3.5 Self Test Project self_test

1. Create a Self-Test Project — create a directory called **`C:\SDK_Demo\self_test`**. Open Unity by selecting Programs -> Ubicom -> Unity from the Windows Start menu. Then, select Project -> New from the Unity menu bar and navigate to the new directory. For the project name, enter **`self_test.c_c`**. For the project type, select SDK, and click the OK button. Select template **`self_test`**, then click the OK button.
2. *Configure the Project* — (optional; only to change the default configuration) From the Unity menu bar, select Tools->Configure, which loads the file **`self_test\config\self_test.lpj`** into the Configuration Tool.
   - Generate the header files that pass configuration parameters to the build process (menu Package -> Generate).
   - Close the configuration tool.
3. *Compile Project* — Select Build -> Compile from the Unity menu bar.
4. *Download Project* — Enter device programming mode by selecting the Build -> Start Programmer command from the Unity menu bar. In the programmer window that appears, click the Program button. When the "Complete OK" message appears, click the Close button.
5. *Verify Operation* — This program tests external flash, Ethernet (connected to a hub/switch), WLAN PC-Card, and indirectly the IP2022. It lights LEDs to communicate status.

During the test, LEDs D7, D8 and D9 should flash. After the test, these same 3 LEDs should be on; if not, something is not right, and the LEDs indicate the problem area:

– LED D7 OFF: ethernet link failed
– LED D8 OFF: external serial flash failed
– LED D9 OFF: PCMCIA interface failed

4.0 UI Generator and Elf Editor

4.1 Post-Compile Configuration

The Elf Editor and the UI Generator are tools that aid in changing application parameters at two critial points after the application software has been compiled:

- *Pre-load* — The Elf Editor can be used to set the factory default parameters of the application on a lab PC with MS Windows before programming the IP2022 device.
- *Run-time* — The HTTP/HTML interface created by the UI Generator can be used (by an end user or network engineer, for example) to set local preferences in the Flash memory of the IP2022 device using a web browser on any platform.

This chapter describes how to use the UI Generator to modify the web interfaces of the Wireless LAN (802.11) SDK projects, and also describes the use of the Elf Editor to change the factory default options.

4.2 Master UI Specifications File

The source specifications for user interface (UI) generated by the UI Generator and Elf Editor reside in a text file which is identified by the project name with the extension ".lua". The specifications include what items are configurable, and for each item its:

- Data Type — For exmple: numerated choice, range of numeric values.
- Default Value — This is where factory default values are first defined.
- Presentation — Where and how it should be displayed.

These source specification are processed during the build process by the program **`uigen.exe`** and made available in an encoded format to **`elfaddfile.exe`** and to the Elf Editor program after the build is complete. The Elf Editor can change the factory default values after compilation but before programming the device. The UI Generator creates the HTML and CGI code needed to create a run-time HTML browser interface.

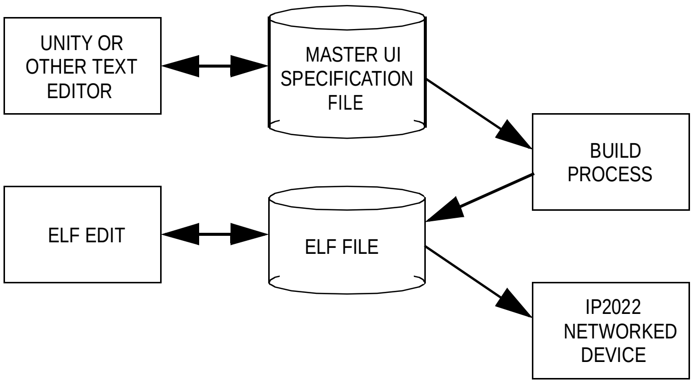


Figure 4-1 Flow of UI Specifications

Examples of Master UI Specification Files are supplied with the example projects. They are stored in the project's **`config`** subdirectory.

The Master UI Specification File is arranged in two main sections: the data section and the web pages structure. The data section describes the configurable data found on the web pages, while the web pages section (also known as “the views”) describes what data goes on each web page.

You can modify the contents of the Master UI Specification File to suit your application. The syntax of the specifications is that of LUA, an embedded scripting language. See www.lua.org for detailed information.

Note: When the UI Generator is being used, the default values of the items in the UI are defined in the data section of the Master UI Specification File; they are not defined by the Configuration Tool.

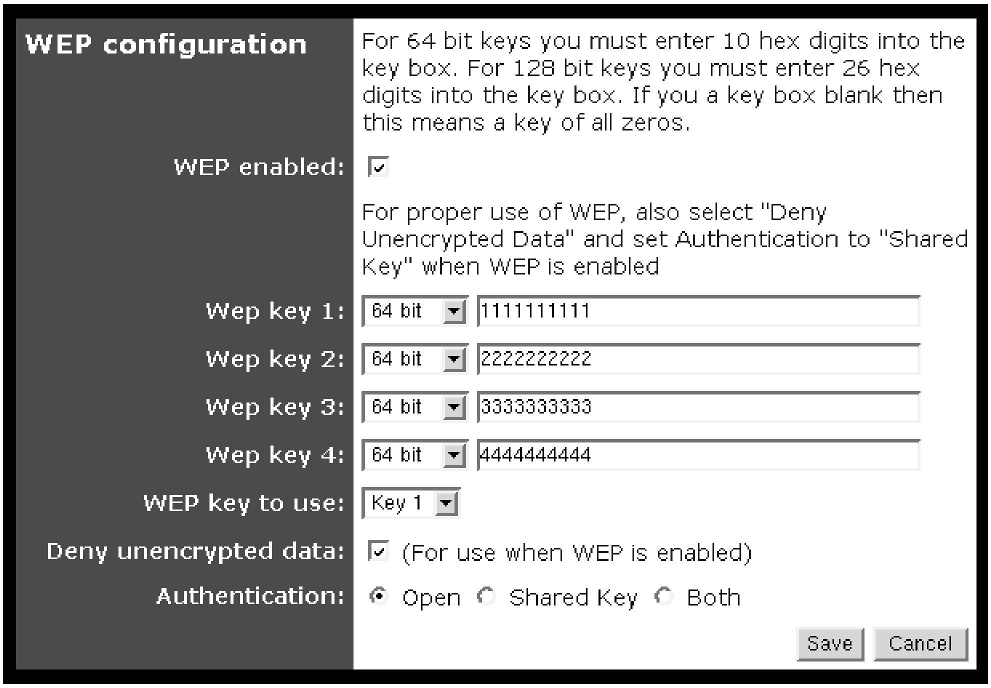
Note specifically that the factory default IP settings are defined here, not in the project’s **`.lpj`** file by the Configuratio Tool. For example (from **`wlan_ap.lua`**):

```
ip_address = ipv4_address {
    info = "Default IP address",
    default = "192.168.1.90",
},
subnet_mask = ipv4_address {
    info = "Default subnet mask",
    default = "255.255.255.0",
},
gateway = ipv4_address {
    info = "Default gateway",
    default = "192.168.1.1",
},
```

The following sections provide examples of two other kinds of modifications.

4.2.1 Combining features

By changing just the Master UI Specifications, without editing HTML or JavaScript code, it is possible to combine two features of a web page into one. As an example, on the security page shown in Figure 4-2, the “deny unencrypted data” checkbox will be combined with “WEP enabled”:



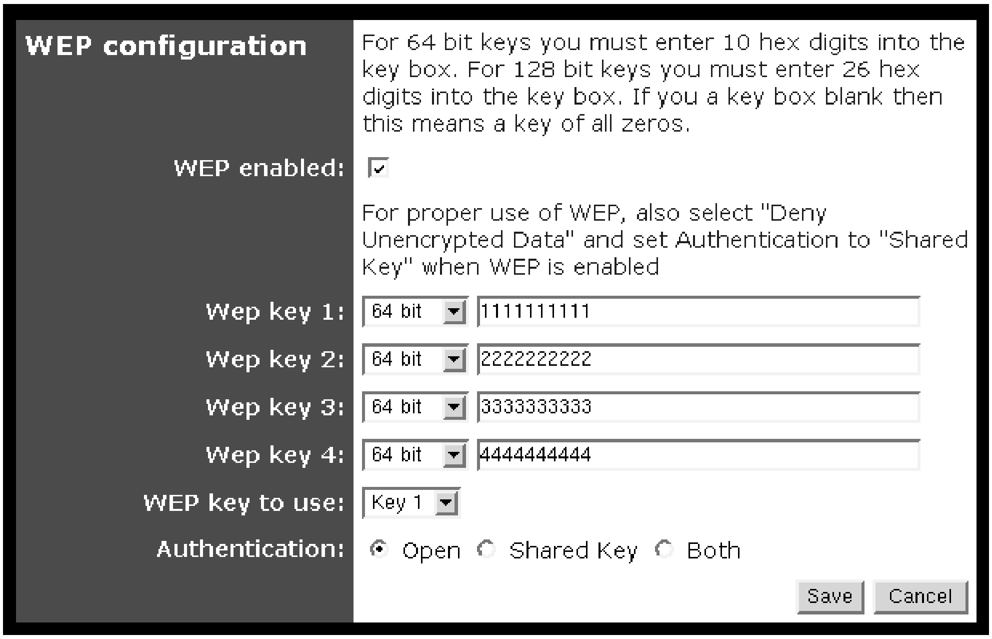
WEP configuration

For 64 bit keys you must enter 10 hex digits into the key box. For 128 bit keys you must enter 26 hex digits into the key box. If you a key box blank then this means a key of all zeros.

WEP enabled:	✓
	For proper use of WEP, also select "Deny Unencrypted Data" and set Authentication to "Shared Key" when WEP is enabled
Wep key 1:	64 bit 1111111111
Wep key 2:	64 bit 2222222222
Wep key 3:	64 bit 3333333333
Wep key 4:	64 bit 4444444444
WEP key to use:	Key 1
Deny unencrypted data:	✓ (For use when WEP is enabled)
Authentication:	[✓ Open] Shared Key Both

Figure 4-2 Web UI Before Combining

In other words, the “deny unencrypted data” checkbox is removed from the page, and the option is in effect when WEP is enabled. Figure 4-3 shows the modified web page.



WEP configuration

For 64 bit keys you must enter 10 hex digits into the key box. For 128 bit keys you must enter 26 hex digits into the key box. If you a key box blank then this means a key of all zeros.

WEP enabled: ✓

For proper use of WEP, also select "Deny Unencrypted Data" and set Authentication to "Shared Key" when WEP is enabled

Wep key 1:	64 bit	1111111111
Wep key 2:	64 bit	2222222222
Wep key 3:	64 bit	3333333333
Wep key 4:	64 bit	4444444444
WEP key to use:	Key 1	

Authentication: [✓ Open] Shared Key Both

Figure 4-3 Web UI After Combining

To remove the "deny unencrypted data" checkbox, the Master UI Specification File must be edited. Open **`config/wlan_ap.lua`** and delete (or comment) the "deny" block in **`the_data`** struture, this removes the corresponding struture element in the C code:

```
the_data = struct {
…
    deny = boolean {
        info = "Deny unencrypted data",
        info2 = "For use when WEP is enabled",
        help = "Select this to require peers to use
encryption",
        default = FALSE,
    },
…
```

Delete also the “deny” item in the_views section. This removes the checkbox from the web page:

```
the_views = {
…
    {
        name = 'Security',
…
            {'use_key'},
            {'deny'},
            {'auth'},
…
```

Next, as the generated C structure will lack the element “deny”, the C code must reflect this change. Open **`app/prism.c`**, look for the function **`write_security_config_to_PRISM`**:

```
void write_security_config_to_PRISM()
{
…
    if (config_v.wepon) {
        x = 1;
        if (config_v.deny) x |= 2;
    }
…
```

The **`config_v.deny`** does not exist anymore, and must be combined with **`config_v.wepon`**, thus we can modify the function like this:

```
void write_security_config_to_PRISM()
{
…
    if (config_v.wepon) {
        x = 3;
    }
…
```

Rebuilding the project regenerates the software and the web pages.

4.2.2 Moving a feature between web pages

Moving a feature from one web page to another only requires changing the Master UI Specification File used by the UI Generator. The data section of the Master UI Specification File does not need to change; only where the feature appears in the views.

As an example, the maximum number of associated stations will be moved from the advanced page to the security page. In **`config/wlan_ap.lua`**, the data is:

```
the_data = struct {
…
    max_assoc = int {
        info = "Maximum associated stations",
        type = "u8",
        default = 200,
        min = 1,
        max = 200,
    },
…
```

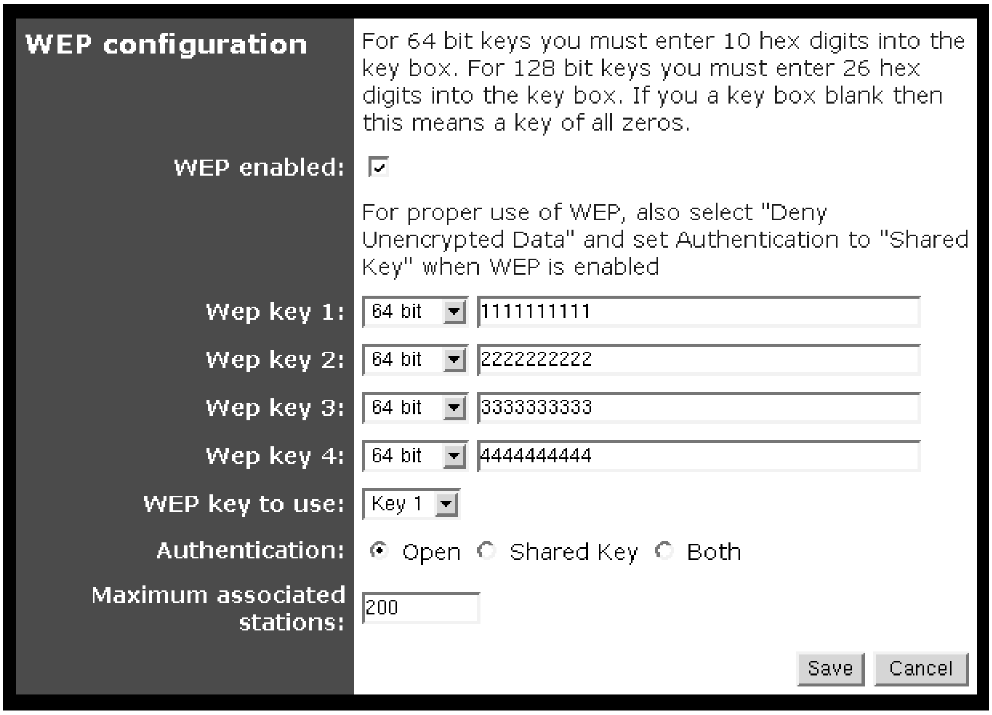
...and the corresponding view is:

```
    items = {
        {'max_assoc'},
        {'exclude'},
        {'frag_threshold'},
        {'RTS_threshold'},
        {'beacon_period'},
        {'dtim_interval'},
        {'multicast_buffering'},
    },
    html_output = webdir..'advanced.html',
```

The item **`max_assoc`** then needs to move:

```
…
        {'wep_key[3].key'; same_line=1},
{'use_key'},
        {'auth'},
        {'max_assoc'},
    },
    html_output = webdir..'security.html',
```

Rebuilding the project generates the web page shown in Figure 4-4.



WEP configuration

For 64 bit keys you must enter 10 hex digits into the key box. For 128 bit keys you must enter 26 hex digits into the key box. If you a key box blank then this means a key of all zeros.

WEP enabled: ✓

For proper use of WEP, also select "Deny Unencrypted Data" and set Authentication to "Shared Key" when WEP is enabled

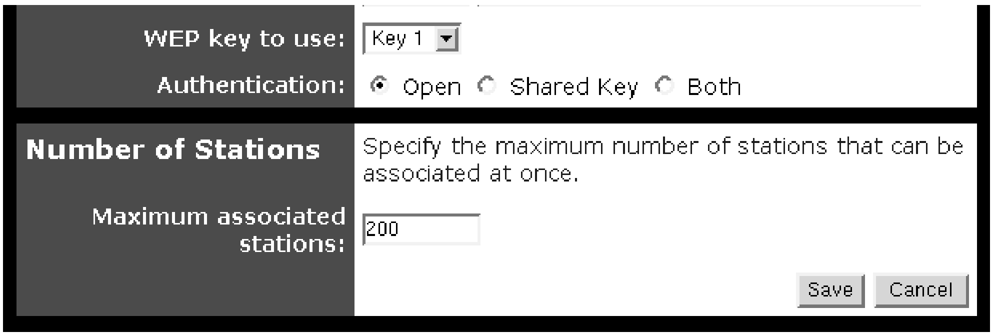
Wep key 1:	64 bit	1111111111
Wep key 2:	64 bit	2222222222
Wep key 3:	64 bit	3333333333
Wep key 4:	64 bit	4444444444
WEP key to use:	Key 1	
Authentication:	[✓ Open] Shared Key Both	
Maximum associated stations:	200	

Figure 4-4 Web UI Before Moving Control

Obviously, this feature is not part of the WEP configuration and should be on its own section on this page:

```
…
        {'auth'},
        {'max_assoc';
            section_name = 'Number of Stations',
            section_help = [[
Specify the maximum number of stations that can be
associated at once.]],
        },
    },
    html_output = webdir..'security.html',
…
```

Now, the maximum number of associated stations is on its own section, as Figure 4-5 shows.



Authentication: [✓ Open] Shared Key Both

Specify the maximum number of stations that can be associated at once.

Maximum associated stations: 200

Figure 4-5 Web UI After Moving Control

4.3 UI Generator

The UI Generator program **`uigen.exe`** is invoked during the build process. The UI Generator interprets the Master UI Specification File and generates the HTML text and the CGI functions (i.e. the C code) that deliver web pages through the IP2022-based HTTP server.

When the UI Generator creates the HTML and CGI files, it also generates a **`ui_config.c`**, **`ui_config.h`**, and a "flattened" configuration file **`flat.lua`** in the project's **`app`** subdirectory. The **`flat.lua`** can be inserted into the ELF file using the program ElfAddFile. The Elf Editor builds its UI dynamically from this symbolic information in the ELF file.

There is a slight size penalty for using the Elf Editor, because ElfAddFile actually adds a dummy function with variables to the ELF file.

Execution of the UI Generator and ElfAddFile is typically specified in the project's makefile. In the example projects provided with the Advanced Wireless Kit, the makefiles are already prepared to use the UI Generator and ElfAddFile, so that the compiled ELF files support the Elf Editor.

4.4 Elf Editor

The Elf Editor program is a Windows based tool, which is used, prior to programming a device, to edit the factory default parameters embedded in IP2022 binary ELF files.

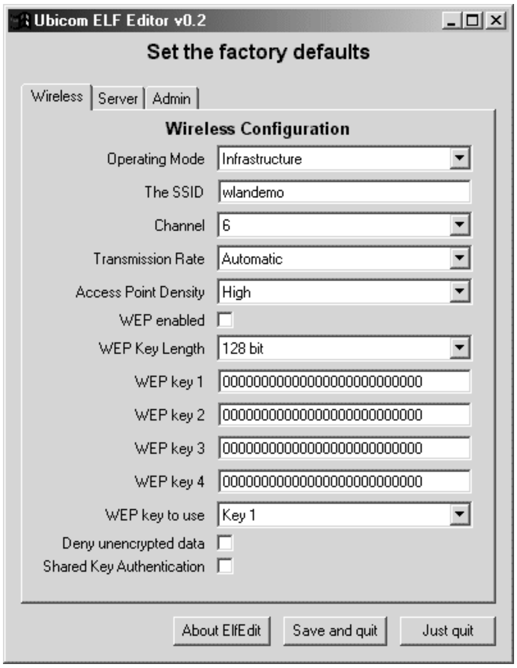
The Elf Editor works only with specially compiled ELF files. The Elf Editor looks for a specific section of the ELF file that specifies the editable parameters. The editable parameters are the same as the ones defined in the Master UI Specification File used by the UI Generator, and are, therefore, the same as the ones found on the web pages.

4.4.1 Using the Elf Editor

The Elf Editor program **`elfedit.exe`** resides in the **`Ubicom\sdk\tools`** directory. To run it double-click on its entry in a Windows directory listing. For convenience you may want to create a shortcut for the Start menu or the Quick Launch section of the Windows Task Bar. The program requires one of the specially compiled ELF files; there are no other dependencies.

Typically, after successfully compiling and testing code, you will want to make a copy of the ELF to be modified. You then open the the Elf Editor for this new copy. A separate copy of the pre-compiled ELF file should be used for each instance in which the factory defaults differ from the original.

Launching the Elf Editor opens a dialog box that prompts for an ELF file. You can also launch the Elf Editor by dragging an ELF file onto a shortcut icon for the Elf Editor. (Selecting an ELF file that does not contain any editable parameters results in an error message.) Once an appropriate ELF file is loaded, the Elf Editor window shows several tabs (see Figure 4-6). Each tab corresponds to one of the web pages generated by the UI Generator, and each pane shows the same parameters and default values that are shown by the corresponding web page.


Ubicom ELF Editor v0.2

Set the factory defaults

Wireless Configuration

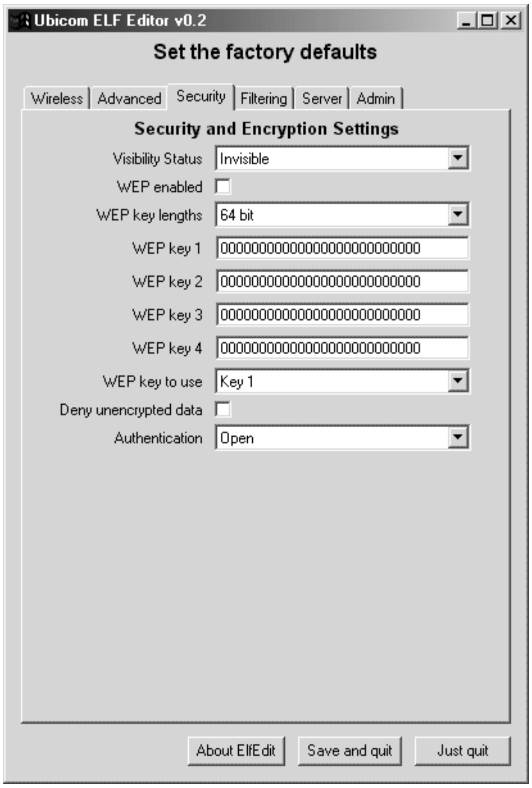
Operating Mode	Infrastructure
The SSID	wlandemo
Channel	6
Transmission Rate	Automatic
Access Point Density	High
WEP enabled	☐
WEP Key Length	128 bit
WEP key 1	00000000000000000000000000
WEP key 2	00000000000000000000000000
WEP key 3	00000000000000000000000000
WEP key 4	00000000000000000000000000
WEP key to use	Key 1
Deny unencrypted data	☐
Shared Key Authentication	☐

Figure 4-6 Elf Editor Window

After editing parameters, you can cancel the changes by clicking the “Just Quit” button, or you can save the changes to the ELF file (make sure you have a backup copy). No boundary check is done on the parameters’ values; so, it is possible to enter values that could cause the application not to work as expected.

4.5 One Specification, Two Presentations

By comparing Figure 4-7 and Figure 4-8 you can see how the web configuration page and the Elf Editor page generated from the same Master UI Specifications present analogous information and controls, but with different presentation formats.

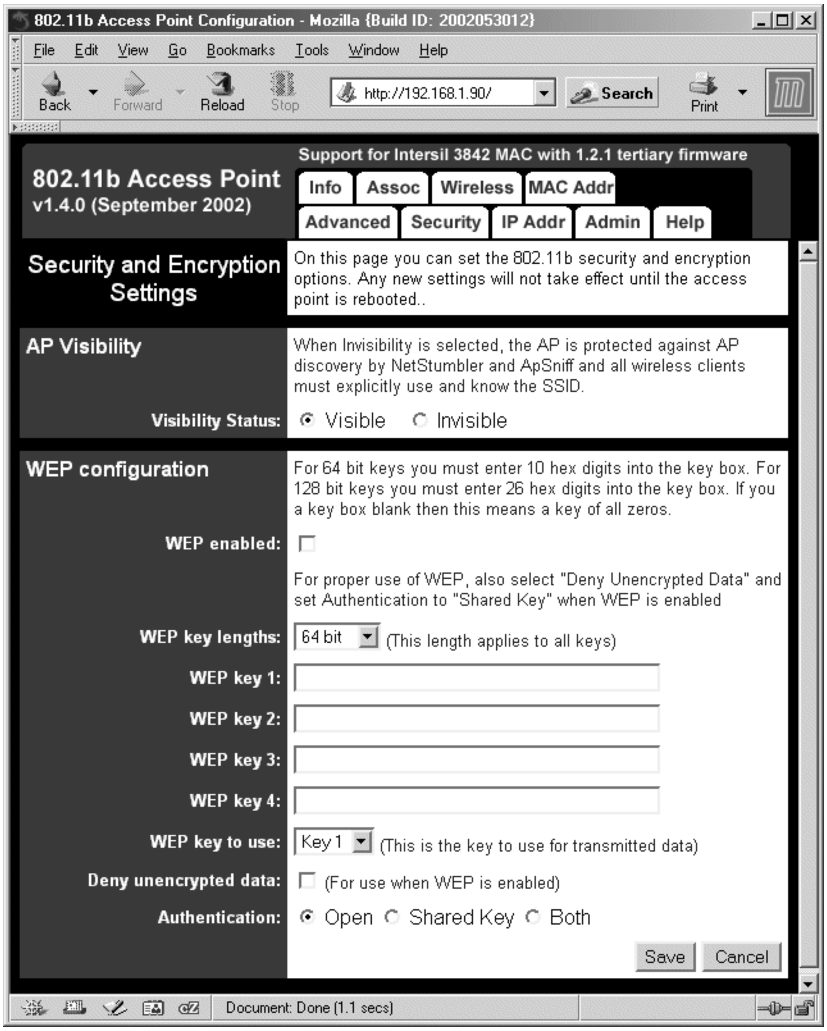


Set the factory defaults

Security and Encryption Settings

Visibility Status	Invisible
WEP enabled	☐
WEP key lengths	64 bit
WEP key 1	00000000000000000000000000
WEP key 2	00000000000000000000000000
WEP key 3	00000000000000000000000000
WEP key 4	00000000000000000000000000
WEP key to use	Key 1
Deny unencrypted data	☐
Authentication	Open

Figure 4-7 Elf Editor UI Example



802.11b Access Point Configuration - Mozilla {Build ID: 2002053012}
File Edit View Go Bookmarks Tools Window Help
Back
Forward
Reload
Stop
http://192.168.1.90/
Search
Print
Info
Assoc
Wireless
MAC Addr
Advanced
Security
IP Addr
Admin
Help
Save
Cancel
Document: Done (1.1 secs)

Support for Intersil 3842 MAC with 1.2.1 tertiary firmware

802.11b Access Point
v1.4.0 (September 2002)

Security and Encryption Settings

On this page you can set the 802.11b security and encryption options. Any new settings will not take effect until the access point is rebooted..

AP Visibility

When Invisibility is selected, the AP is protected against AP discovery by NetStumbler and ApSniff and all wireless clients must explicitly use and know the SSID.

Visibility Status: [✓ Visible] Invisible

WEP configuration

For 64 bit keys you must enter 10 hex digits into the key box. For 128 bit keys you must enter 26 hex digits into the key box. If you a key box blank then this means a key of all zeros.

WEP enabled: ☐

For proper use of WEP, also select "Deny Unencrypted Data" and set Authentication to "Shared Key" when WEP is enabled

WEP key lengths: 64 bit (This length applies to all keys)

WEP key 1:

WEP key 2:

WEP key 3:

WEP key 4:

WEP key to use: Key 1 (This is the key to use for transmitted data)

Deny unencrypted data: ☐ (For use when WEP is enabled)

Authentication: [✓ Open] Shared Key Both

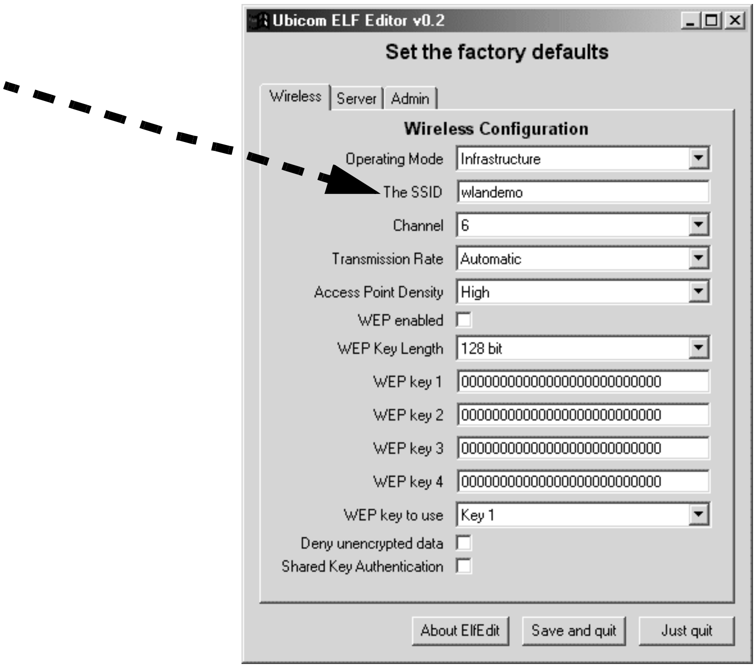
Figure 4-8 Web Configuration UI Example

5.0

Modifying Example Projects

Referring to the Access Point and Bridge projects of Chapter 3, this chapter identifies some of the application parameters most likely to be changed and points out where to change them.

5.1 SSID (factory default)

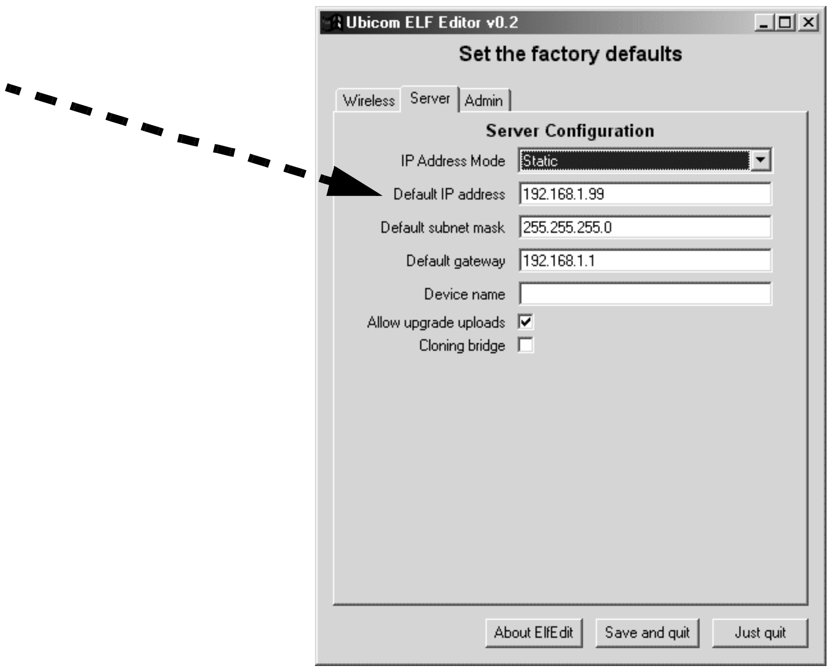


Set the factory defaults

Wireless Configuration

Operating Mode	Infrastructure
The SSID	wlandemo
Channel	6
Transmission Rate	Automatic
Access Point Density	High
WEP enabled	☐
WEP Key Length	128 bit
WEP key 1	00000000000000000000000000
WEP key 2	00000000000000000000000000
WEP key 3	00000000000000000000000000
WEP key 4	00000000000000000000000000
WEP key to use	Key 1
Deny unencrypted data	☐
Shared Key Authentication	☐

5.2 IP Address (factory default)

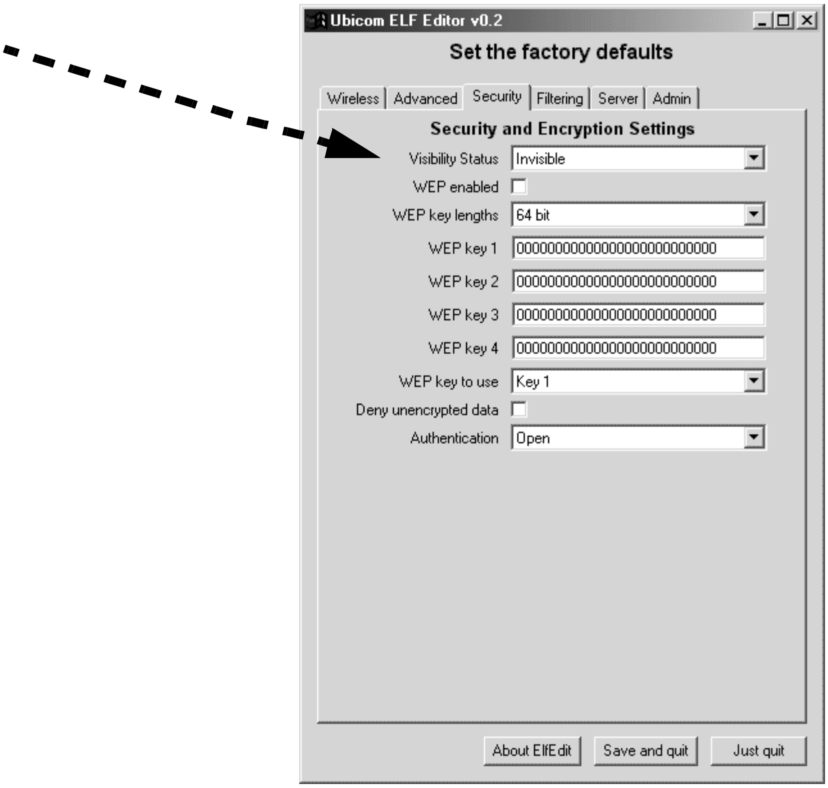


Set the factory defaults

Server Configuration

IP Address Mode	Static
Default IP address	192.168.1.99
Default subnet mask	255.255.255.0
Default gateway	192.168.1.1
Device name	
Allow upgrade uploads	✓
Cloning bridge	☐

5.3 AP Visibility (factory default)



Set the factory defaults

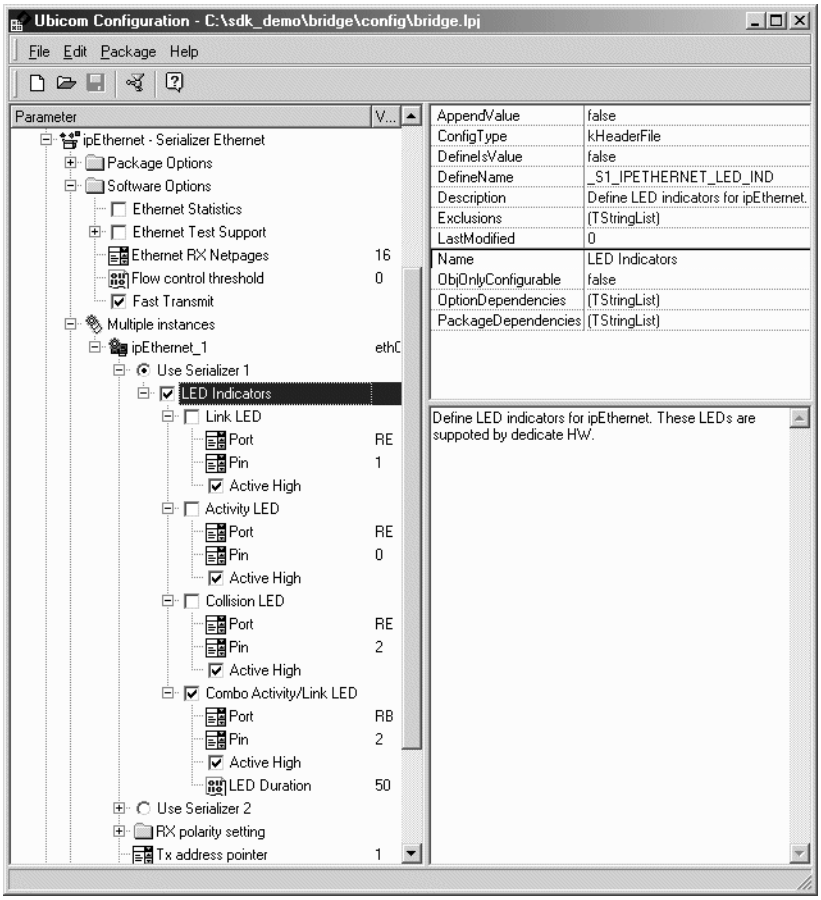
Security and Encryption Settings

Visibility Status	Invisible
WEP enabled	☐
WEP key lengths	64 bit
WEP key 1	00000000000000000000000000
WEP key 2	00000000000000000000000000
WEP key 3	00000000000000000000000000
WEP key 4	00000000000000000000000000
WEP key to use	Key 1
Deny unencrypted data	☐
Authentication	Open

5.4 LED Assignments and Functions

Some designers will choose to use a single WLAN LED. To do so, uncheck the WLAN Tx and WLAN Rx LEDs, and select the WLAN

Combo LED, which will remain solid to indicate a WLAN connection and will flash to indicate Rx to Tx WLAN traffic.



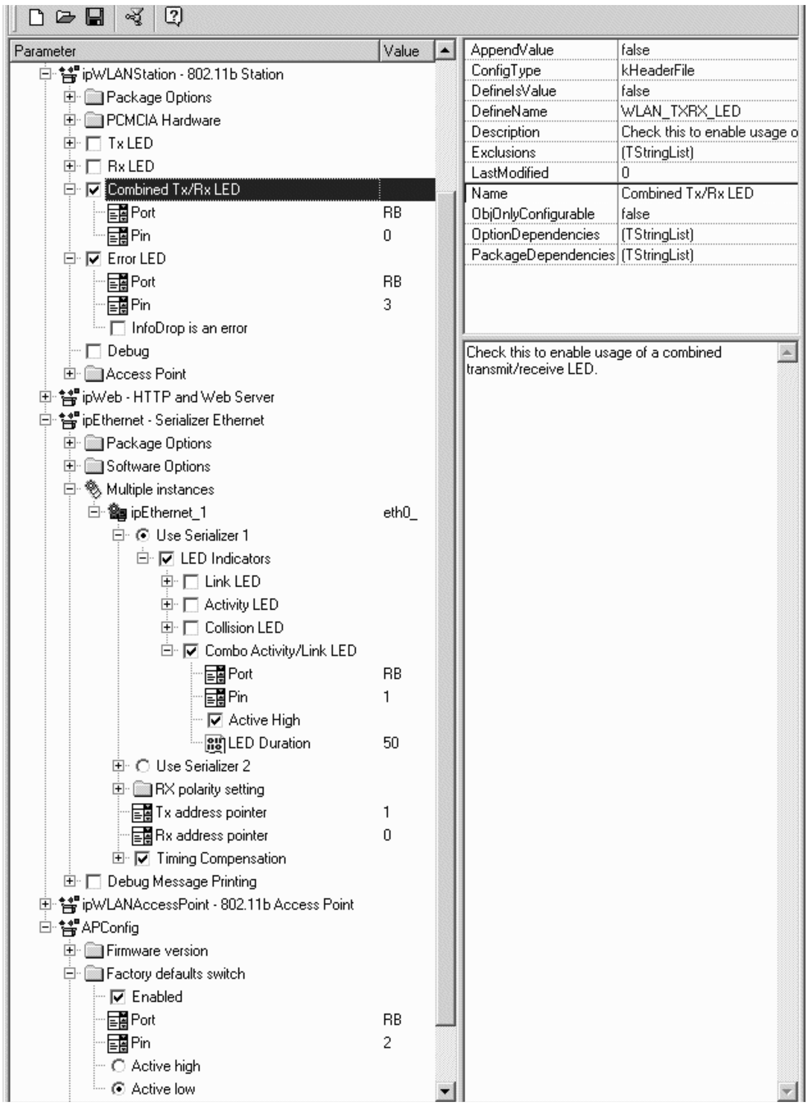
Parameter	V...
ipEthernet - Serializer Ethernet	
Package Options	
Software Options	
☐ Ethernet Statistics	
☐ Ethernet Test Support	
Ethernet RX Netpages	16
Flow control threshold	0
☑ Fast Transmit	
Multiple instances	
ipEthernet_1	eth(
● Use Serializer 1	
☑ LED Indicators	
☐ Link LED	
Port	RE
Pin	1
☑ Active High	
☐ Activity LED	
Port	RE
Pin	0
☑ Active High	
☐ Collision LED	
Port	RE
Pin	2
☑ Active High	
☑ Combo Activity/Link LED	
Port	RB
Pin	2
☑ Active High	
LED Duration	50
○ Use Serializer 2	
RX polarity setting	
Tx address pointer	1

AppendValue	false
ConfigType	kHeaderFile
DefineIsValue	false
DefineName	_S1_IPETHERNET_LED_IND
Description	Define LED indicators for ipEthernet.
Exclusions	(TStringList)
LastModified	0
Name	LED Indicators
ObjOnlyConfigurable	false
OptionDependencies	(TStringList)
PackageDependencies	(TStringList)

Define LED indicators for ipEthernet. These LEDs are suppoted by dedicate HW.

5.5 Reset to Factory Default switch (RB2)

Note that the SW2 switch on the IP2022 Demo Board v3.0 is hardwired to RB2, which is also the D6 LED, which in these WLAN projects is used for Ethernet. To use SW2 as the “Reset to Factory Defaults” switch, use the Config Tool to make RB0 (D8) WLAN combo, RB1 (D7) Ethernet Combo, and keep RB3 (D5) as Error/Diagnostic. Then set RB2, active low logic.



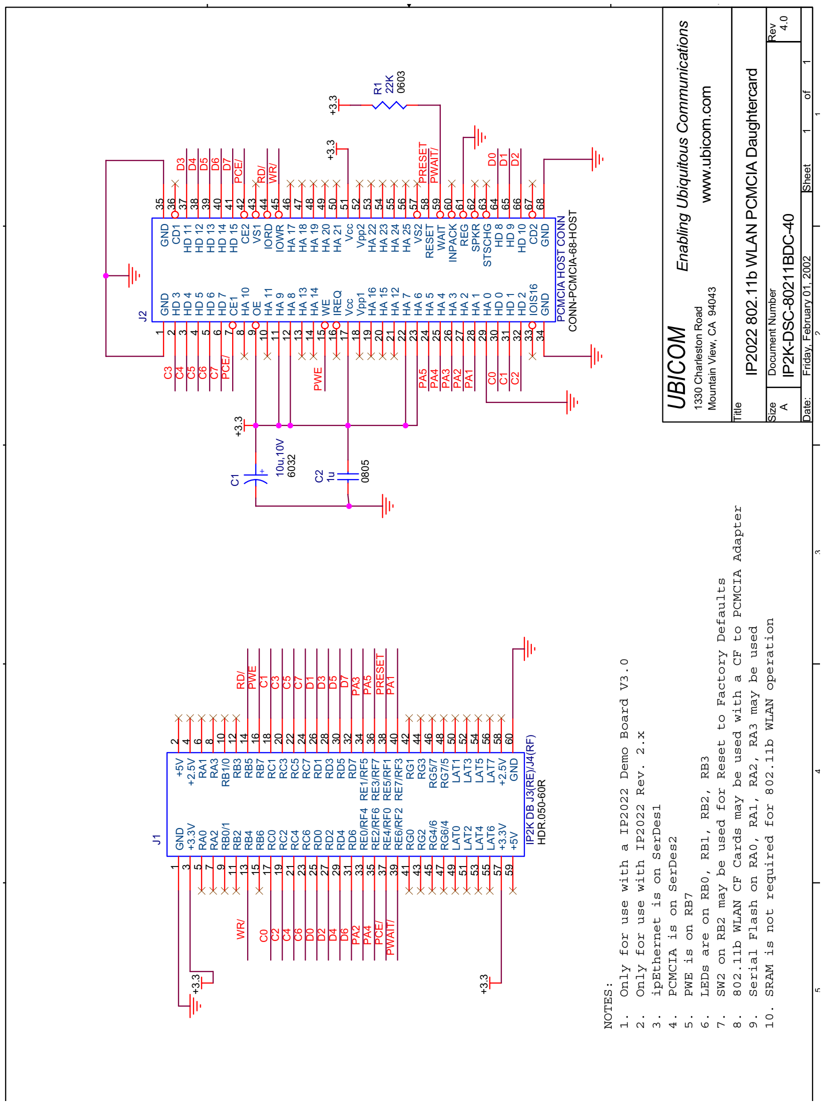
```
Parameter                                         Value
ipWLANStation - 802.11b Station
    Package Options
    PCMCIA Hardware
    [ ] Tx LED
    [ ] Rx LED
    [✓] Combined Tx/Rx LED
        Port                                      RB
        Pin                                       0
    [✓] Error LED
        Port                                      RB
        Pin                                       3
        [ ] InfoDrop is an error
    [ ] Debug
    Access Point
ipWeb - HTTP and Web Server
ipEthernet - Serializer Ethernet
    Package Options
    Software Options
    Multiple instances
        ipEthernet_1                              eth0_
            (●) Use Serializer 1
                [✓] LED Indicators
                    [ ] Link LED
                    [ ] Activity LED
                    [ ] Collision LED
                    [✓] Combo Activity/Link LED
                        Port                      RB
                        Pin                       1
                        [✓] Active High
                        LED Duration              50
            ( ) Use Serializer 2
            RX polarity setting
            Tx address pointer                    1
            Rx address pointer                    0
            [✓] Timing Compensation
    [ ] Debug Message Printing
ipWLANAccessPoint - 802.11b Access Point
APConfig
    Firmware version
    Factory defaults switch
        [✓] Enabled
        Port                                      RB
        Pin                                       2
        ( ) Active high
        (●) Active low
```

AppendValue	false
ConfigType	kHeaderFile
DefineIsValue	false
DefineName	WLAN_TXRX_LED
Description	Check this to enable usage o
Exclusions	(TStringList)
LastModified	0
Name	Combined Tx/Rx LED
ObjOnlyConfigurable	false
OptionDependencies	(TStringList)
PackageDependencies	(TStringList)

Check this to enable usage of a combined transmit/receive LED.

A.0 Schematics

A.1 Advanced Wireless Kit Daughtercard



```
NOTES:
1.  Only for use with a IP2022 Demo Board V3.0
2.  Only for use with IP2022 Rev. 2.x
3.  ipEthernet is on SerDes1
4.  PCMCIA is on SerDes2
5.  PWE is on RB7
6.  LEDs are on RB0, RB1, RB2, RB3
7.  SW2 on RB2 may be used for Reset to Factory Defaults
8.  802.11b WLAN CF Cards may be used with a CF to PCMCIA Adapter
9.  Serial Flash on RA0, RA1, RA2, RA3 may be used
10. SRAM is not required for 802.11b WLAN operation
```