Software User Manual

Unmanaged

Getting Started Guide

RELEASE UM-3.4.0

Unmanaged-SWUM104-R

Corporate Headquarters: San Jose, CA

May 19, 2016

10/12/2016 VRWTC

Revision History

Revision	*Date*	*Change Description*
Unmanaged-SWUM104-R	05/19/16	Document updated for the 3.4.0 release.
Unmanaged-SWUM103-R	01/19/16	Document updated for the 3.3.X release.
Unmanaged-SWUM102-R	07/30/15	Document updated for the 3.3.0 release.
Unmanaged-SWUM101-R	02/23/15	Document updated for the 3.2.2 release.
Unmanaged-SWUM100-R	12/18/14	Initial release

Table of Contents

List of Figures

List of Tables

About This Document

Purpose and Audience

非受控

Broadcom® Unmanaged Software is available in three different packages: Unmanaged Web (UM-Web), Unmanaged Plus (UM+), and Unmanaged Dumb (UM-Dumb). This document applies to all packages.

不说话

Acronyms and Abbreviations

In most cases, acronyms and abbreviations are defined on first use.

For a comprehensive list of acronyms and other terms used in Broadcom documents, go to: http://www.broadcom.com/press/glossary.php.

Document Conventions

The following conventions may be used in this document:

Convention	*Description*
Bold	User input and actions: for example, type **exit**, click **OK**, press **Alt+C**
`Monospace`	Code: `#include <iostream>` HTML: `<td rowspan = 3>` Command line commands and parameters: `wl [-l] <command>`
< >	Placeholders for *required* elements: enter your <username> or `wl <command>`
[ ]	Indicates *optional* command-line parameters: `wl [-l]` Indicates bit and byte ranges (inclusive): [0:3] or [7:0]

References

The references in this section may be used in conjunction with this document.

Note: Broadcom provides customer access to technical documentation and software through its Customer Support Portal (CSP) and Downloads and Support site (see Technical Support).
客户支持门户

For Broadcom documents, replace the “xx” in the document number with the largest number available in the repository to ensure that you have the most current version of the document.

Document (or Item) Name	*Number*	*Source*
Broadcom Items		
[1] Unmanaged Software Porting Guide	Unmanaged-PG1*xx*-R	Broadcom CSP

Technical Support

Broadcom provides customer access to a wide range of information, including technical documentation, schematic diagrams, product bill of materials, PCB layout information, and software updates through its customer support portal (https://support.broadcom.com). For a CSP account, contact your Sales or Engineering support representative.

In addition, Broadcom provides other product support through its Downloads and Support site (http://www.broadcom.com/support/).

Introduction

Broadcom's Unmanaged Software is available in three different packages: Unmanaged Web (UM-Web), Unmanaged Plus (UM+) and Unmanaged Dumb (UM-Dumb). This document applies to all packages.

Purpose

- Add on value of unmanaged device
- Leverage embedded resources of silicon
  - Processor
  - SRAM
- Less cost

Summary

System Services

- Single threaded
  - No OS, no interrupt used
  - CPU timer used as system tick
  - One background task to run all registered tasks
  - SAL layer for each platform
- Start-up code and BSP
  - CPU initialization (cache, MMU/MPU, timer, UART, etc.)
  - Serial flash driver

Switch Management

- Switch init sequence
  - Chip reset and IP/EP/MMU initializations
  - DMA init for packet TX/RX on CPU port
- MAC/PHY drivers
  - Port SDK drivers and remove unnecessary configurations
  - MAC/PHY (SerDes) init sequence at start-up
  - Use MDK PHY library as PHY driver
- Linkscan task
  - Poll PHY status and program port information (speed, duplex, and pause) to MAC according to the link status when the link is up
  - Link down process for MAC/PHY
- Board API for each switch feature
  - For example, `board_vlan_crate()` to create a VLAN
- Includes uIP stack in release package

Functions

UM-Dumb	Only does the basic initialization for CPU/MAC/SerDes/PHY to allow the packets to be switched among the front ports. The purpose of the UM-Dumb package is to let our switch behave as a dumb switch.
UM-Web	Include more L2 features besides the basic initialization of the UM-Dump package. The user can also enable/disable those features through the web page. The firmware upgrade can also be done through the web page.
UM+	Remove WEB features and revise VLAN default settings from the UM-Web package.

Table 1 shows the features for the three packages.

Table 1: Features

Feature Support	*Description*	*UM-Web*	*UM+*	*UM-Dumb*	*Default*
Console/UART	–	V	V	V	9600 bps
L2 AGE	–	V	V	V	300 seconds
EEE	–	V	V	V	Enabled
Jumbo Frame	–	V	V	V	9216 bytes
Port Flow Control	–	V	V	V	Enabled
DoS	Auto Denial of Service prevention. Check DOS_CONTROL and DOS_CONTROL_2 registers for details.	V	V	–	Enabled
IEEE 802.1p QoS	Schedule mode is Weighted Round Robin (WRR) and weighted for priority queues (0, 1, 2, and 3) is (1:2:4:8).	V	V	V	Enabled
DHCP	Note[a]	V	V	–	Enabled
VLAN	–	V	V	V	Note[b]
LAG	Maximum of four groups. Maximum eight ports per group.	V	V	–	Disabled
Mirror	–	V	V	–	Disabled
Port-Based QoS	–	V	V	–	Disabled
Rate Limit	–	V	V	–	Disabled
Storm Control	–	V	V	–	Disabled
IGMP Snooping	–	V	V	–	Enabled
Block Unknown Multicast Address	–	V	V	–	Disabled
Access Control	Covers “limit designated SIP access” and “limit single user access”.	V	V	–	Disabled
Loop Detection	–	V	V	–	Disabled
Cable Diagnostics	–	V	V	–	Triggered by user

Table 1: Features (Cont.)

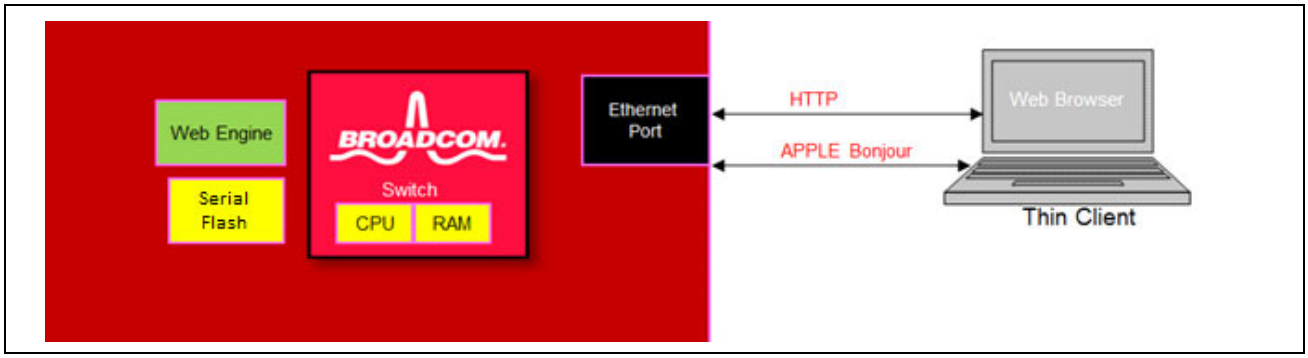
Feature Support	*Description*	*UM-Web*	*UM+*	*UM-Dumb*	*Default*
Port Status	Covers link status, speed duplex, flow control, and auto-negotiation status.	V	–	–	–
Statistics	Covers TX octets, RX octets, and CRC errors.	V	–	–	–
LED Display	Link/Activity LED	V	V	V	–
Firmware Upgrade	–	V	–	–	–
Password Protection	–	V	–	–	–
Snake Test	Support snake test by CLI command.	V	V	V	Triggered by user
Dual Image	–	V	–	–	Enabled
Bonjour	Supports Bonjour browser to discover device.	V	–	–	Enabled
Web	Support web browser to access the device.	V	–	–	–
Persistence	Supports persistence storage.	V	V	–	–
Vendor Config	Configure different settings without rebuilding the image	V	V	V	–

a. For UM-Web, the address is set to a random IPv4 link-local address if DHCP fails. The range of IPv4 Link-local addresses is 169.254.1.0 to 169.254.254.255. For UM+, the address is set to 192.168.0.239 if DHCP fails.

b. For UM-Web, the default setting is 802.1Q VLAN mode. For UM+ and UM-Dumb, the default setting is Port-based VLAN mode. It creates 1 ~ 4094 VLANs as below and sets PVID of all ports to 1:
VLAN 1: All ports in this VLAN are untagged members. 无标签的帧
VLAN 2 ~ 4094: All ports in this VLAN are tagged members. 有标签的帧

Management Mechanisms

You can use a web browser through HTTP to connect to the web page of a device or use Apple Bonjour to discover the device.



Code Directory

The code directory is shown in the following figure.

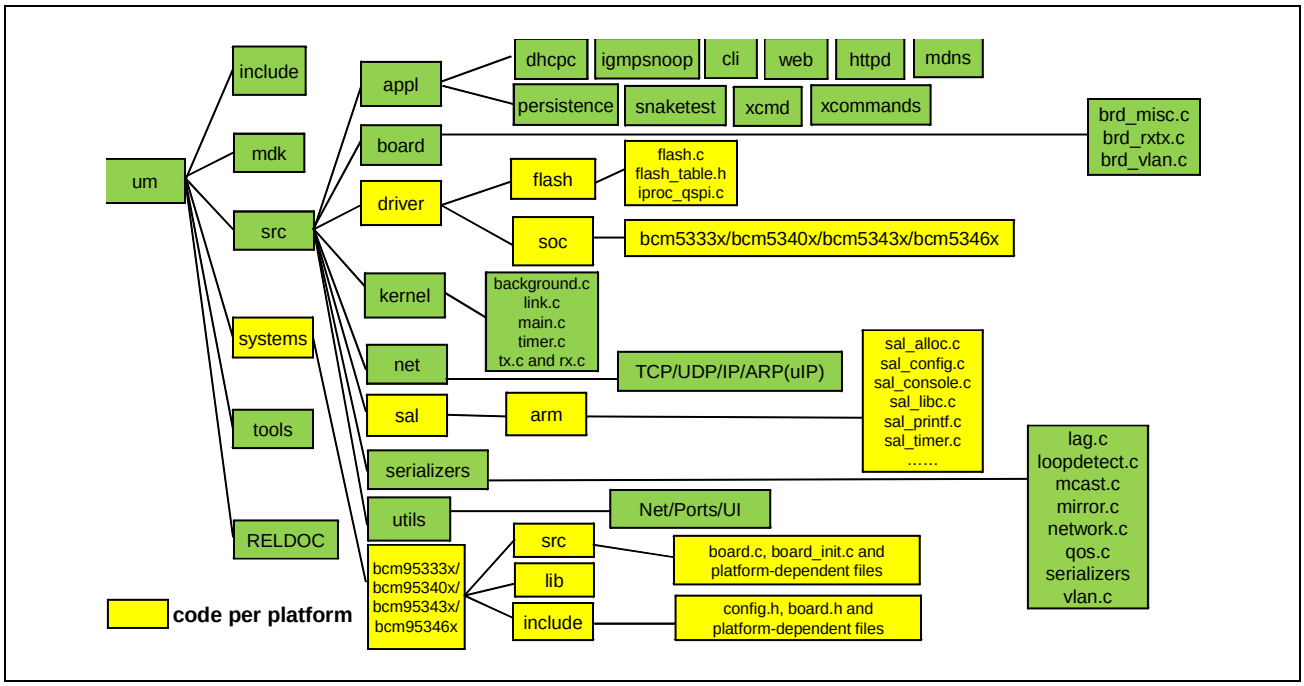


Table 2 gives a brief description of the files that make up the UM code.

Table 2: UM Files

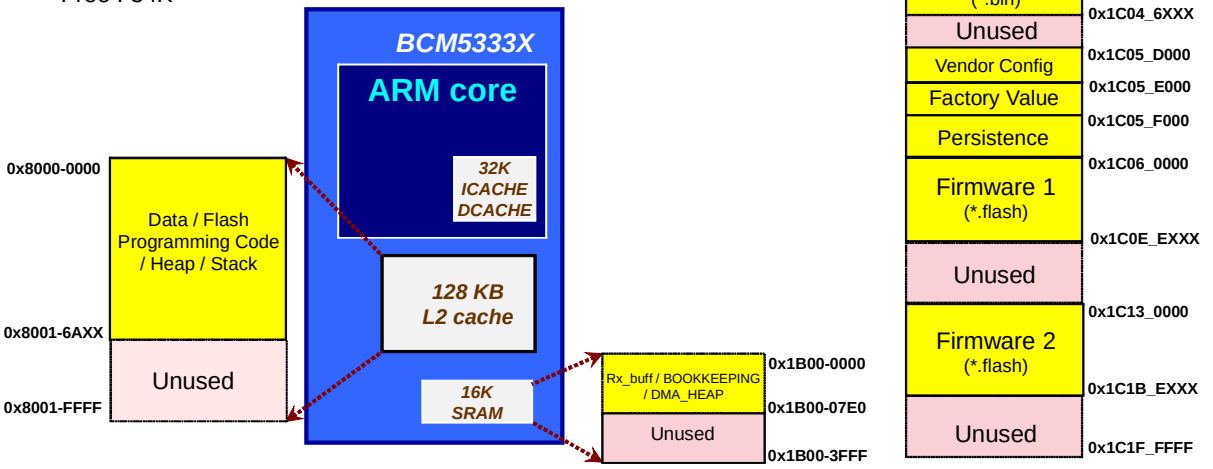
File	Description
$UM/src/appl	Application directory that includes DHCP, IGMPSNOOP, Cisco-like CLI, persistence, HTTPD, MDNS, and debugging CLI.
$UM/src/board	The generic code of the board API is put under this directory.
$UM/src/driver	Includes switch, serial flash driver, and interface code with MDK PHY driver.
$UM/src/kernel	The background task is in this directory. It is the main loop that runs all registered tasks (TX/RX, Timer, and uIP task). Also, the link-scan handler is registered in the timer task and wakes up per 100 ms. The link-change handler is registered in the timer task and wakes up per 600 ms.
$UM/src/net	The uIP code is contained in this directory.
$UM/src/utils	Includes net, port, and debugging CLI utility.
$UM/src/sal	The SAL directory that includes printf, timer, assertion, memory allocation, C libraries, and the console API.
$UM/src/serializers	Includes the serializer code of all features.
$UM/systems	The “BSP source code” and “implementation code of board API” are in this directory. Every board has its own directory under the system directory. The user builds the firmware image under system/per-platform-directory. For example, the user can build a BCM5333X image under *system/bcm95333x* for the BCM5333X platform. After UM 3.1.1, the MDK PHY driver is used in UM. The MDK PHY driver is a library file under *system/per-platform-directory/lib* for each platform.
$UM/mdk	A directory for MDK PHY source code. The user can rebuild the MDK PHY library based on the source code.
$UM/tools	A directory for Perl script, LED-related tools, and assembly code.
$UM/RELDOC	A directory for release documents.

Memory Layout

This section describes the flash and memory layout for UM-Web and UM+ on different platforms.

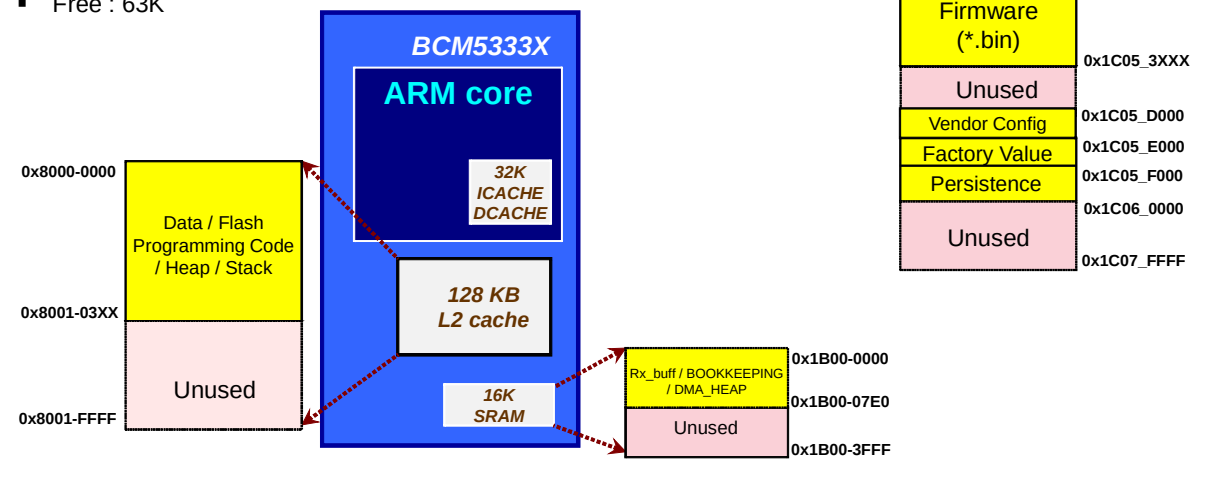
Flash Partitions/Memory Layout for UM-Web (BCM5333X)

- Total Memory Available : 128KB (L2 Cache)
  - Used: 94K
  - Free : 34K

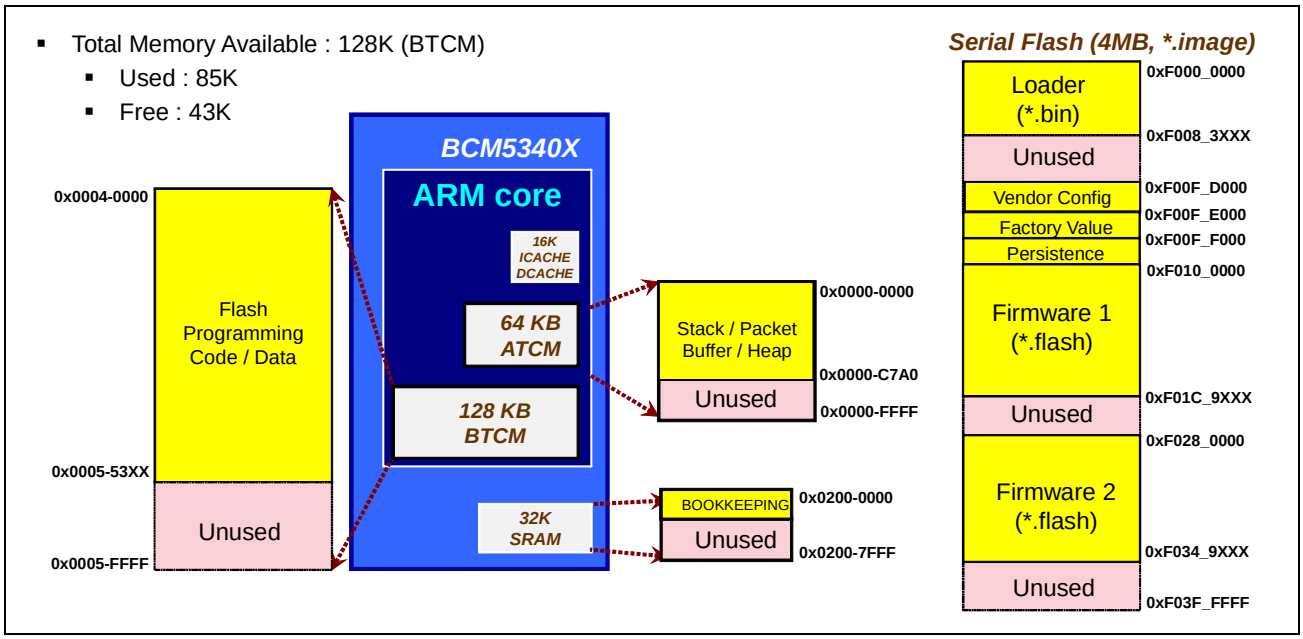


Flash Partitions/Memory Layout for UM+ (BCM5333X)

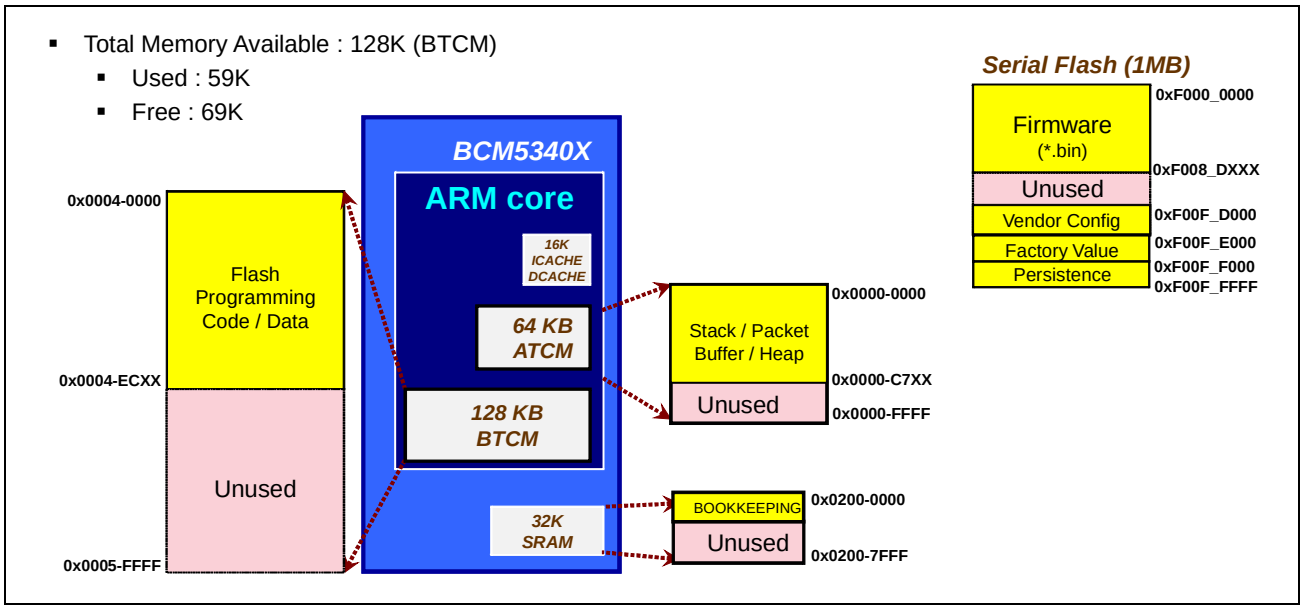
- Total Memory Available : 128K (L2 Cache)
  - Used: 65K
  - Free : 63K



Flash Partitions/Memory Layout for UM-Web (BCM5340X)

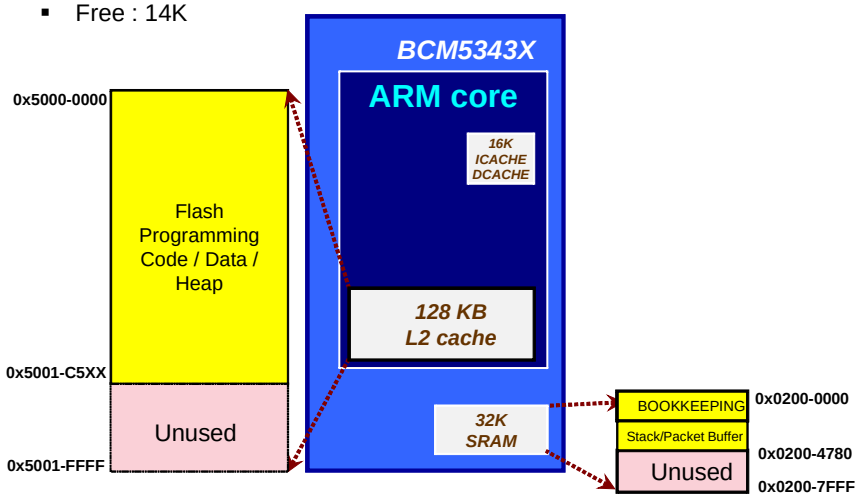


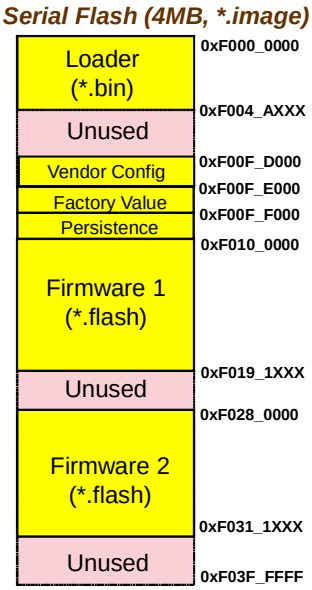
Flash Partitions/Memory Layout for UM+ (BCM5340X)



Flash Partitions/Memory Layout for UM-Web (BCM5343X)

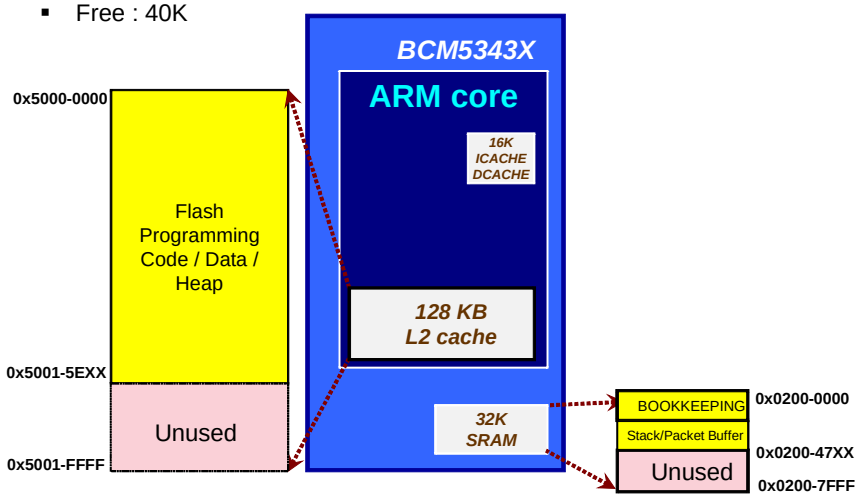
- Total Memory Available : 128K (L2 cache)
  - Used : 114K
  - Free : 14K

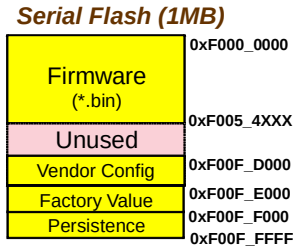




Flash Partitions/Memory Layout for UM+ (BCM5343X)

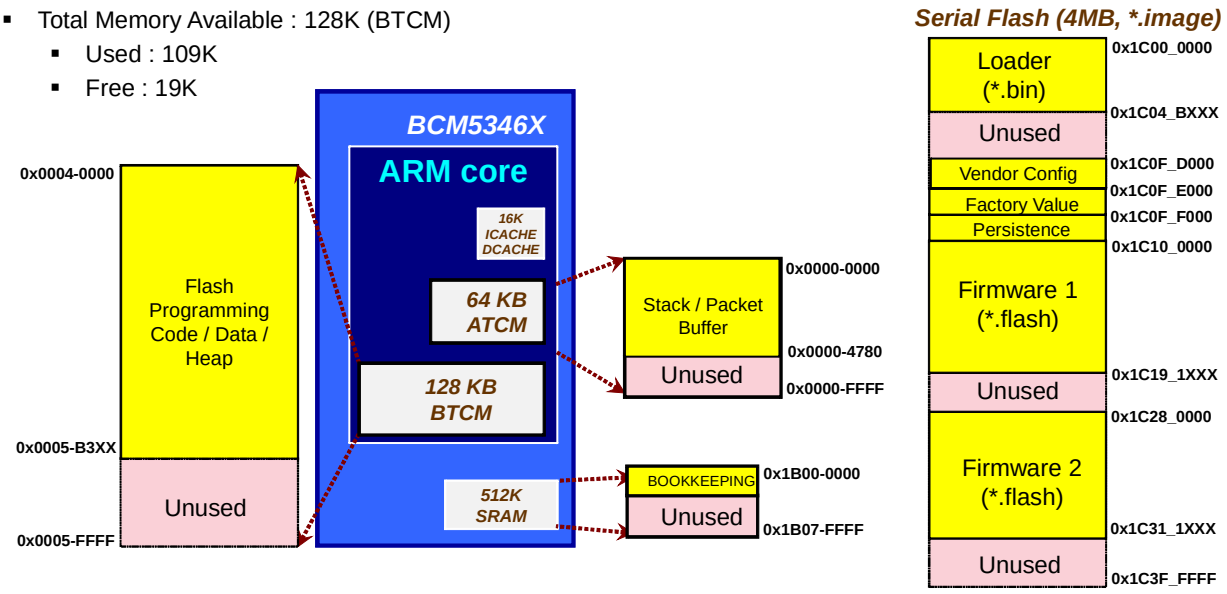
- Total Memory Available : 128K (L2 cache)
  - Used : 88K
  - Free : 40K





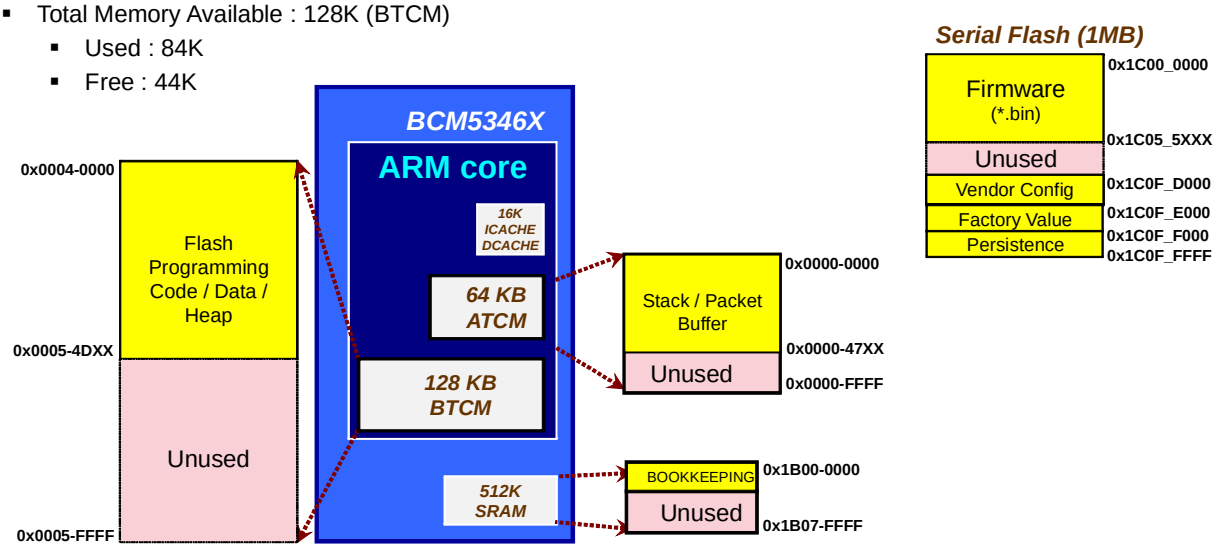
Flash Partitions/Memory Layout for UM-Web (BCM5346X)

- Total Memory Available : 128K (BTCM)
  - Used : 109K
  - Free : 19K



Flash Partitions/Memory Layout for UM+ (BCM5346X)

- Total Memory Available : 128K (BTCM)
  - Used : 84K
  - Free : 44K



Building Steps

Get the source tree in the release tar file.

Toolchain

The BCM5333X, BCM5340X, BCM5343X, and BCM5346X platforms use the same toolchain.

1. Download the GNU GCC ARM Embedded 4.8 update, and install it.
   https://launchpad.net/gcc-arm-embedded/4.8/4.8-2014-q1-update
2. Change TOOLCHAIN_DIR and TOOLPREFIX in *$UM/systems/$PLATFORM/src/tools.mk* to the location where the toolchain is installed. *$PLATFORM* could be bcm95333x, bcm95340x, bcm95343x, or bcm95346x.

Build the MDK PHY Library

修改UM/UK里面的内容需要重新编译PHY库

You can rebuild the MDK PHY library before building the image if you change the PHY code under *um/mdk*.

1. Go to the directory *$UM/systems/$PLATFORM*. *$PLATFORM* could be bcm95333x, bcm95340x, bcm95343x, or bcm95346x.
2. To generate MDK PHY library under *$UM/systems/$PLATFORM/lib*. *$PLATFORM* could be bcm95333x, bcm95340x, bcm95343x, or bcm95346x. Type the following:

```
make phylibs
```

这个文件不用在lib文件夹下执行还是在芯片下执行

Build the Image File for UM-Web

BCM5333X Platform

1. Go to the *$UM/systems/bcm95333x* directory, and create an output directory (outputs).
   ```
   mkdir outputs
   ```
2. Type the following to build the bootloader (*bcm95333x-loader.bin*), and save the bootloader to the output directory.
   ```
   make clean; make target=loader
   cp bcm95333x-loader.bin outputs/
   ```
3. Type the following to build the firmware (*bcm95333x-umweb.flash*), and save the firmware to the output directory.
   ```
   make clean; make target=umweb
   cp bcm95333x-umweb.flash outputs/
   ```
4. Go to output directory and generate a single image (bcm95333x.image) for the BCM5333X platform.
   ```
   cd outputs
   ../../../tools/mkflashimage.pl bcm95333x-loader.bin bcm95333x-umweb.flash bcm95333x.image
   ```
5. Use `flash programmer` to program bcm95333x.image into the flash. Currently, the BCM953334K and BCM953394K use Micro N25Q256A13 serial flash.

BCM5340X Platform

1. Go to the *$UM/systems/bcm95340x* directory, and create an output directory (outputs).
   ```
   mkdir outputs
   ```
2. Type the following to build the bootloader (*bcm95340x-loader.bin*), and save the bootloader to the output directory.
   ```
   make clean; make target=loader
   cp bcm95340x-loader.bin outputs/
   ```
3. Type the following to build the firmware (*bcm95340x-umweb.flash*), and save the firmware to the output directory.
   ```
   make clean; make target=umweb
   cp bcm95340x-umweb.flash outputs/
   ```
4. Go to output directory and generate a single image (*bcm95340x.image*) for the BCM5340X platform.
   ```
   cd outputs
   ../../../tools/mkflashimage.pl bcm95340x-loader.bin bcm95340x-umweb.flash bcm95340x.image
   ```
5. Use `flash programmer` to program *bcm95340x.image* into the flash. Currently, the BCM953406K and BCM953456K use Micro N25Q512A83 serial flash.

BCM5343X Platform

1. Go to the *$UM/systems/bcm95343x* directory, and create an output directory (outputs).
   ```
   mkdir outputs
   ```
2. Type the following to build the bootloader (*bcm95343x-loader.bin*), and save the bootloader to the output directory.
   ```
   make clean; make target=loader
   cp bcm95343x-loader.bin outputs/
   ```
3. Type the following to build the firmware (*bcm95343x-umweb.flash*), and save the firmware to the output directory.
   ```
   make clean; make target=umweb
   cp bcm95343x-umweb.flash outputs/
   ```
4. Go to the output directory and generate a single image (*bcm95343x.image*) for the BCM5343X platform.
   ```
   cd outputs
   ../../../tools/mkflashimage.pl bcm95343x-loader.bin bcm95343x-umweb.flash
   bcm95343x.image
   ```
5. Use flash programmer to program *bcm95343x.image* into the flash. Currently, the BCM953434K uses Winbond W25Q64FV serial flash.

BCM5346X Platform

1. Go to the *$UM/systems/bcm95346x* directory, and create an output directory (outputs).
   ```
   mkdir outputs
   ```
2. Type the following to build the bootloader (*bcm95346x-loader.bin*), and save the bootloader to the output directory.
   ```
   make clean; make target=loader
   cp bcm95346x-loader.bin outputs/
   ```
3. Type the following to build the firmware (*bcm95346x-umweb.flash*), and save the firmware to the output directory.
   ```
   make clean; make target=umweb
   cp bcm95346x-umweb.flash outputs/
   ```
4. Go to the output directory and generate a single image (*bcm95346x.image*) for the BCM5346X platform.
   ```
   cd outputs
   ../../../tools/mkflashimage.pl bcm95346x-loader.bin bcm95346x-umweb.flash
   bcm95346x.image
   ```
5. Use flash programmer to program *bcm95346x.image* into the flash. Currently, the BCM956270K uses Macronix MX25L12835FMI-10G serial flash.

Note: Files with the extension **.image* are used for the flash programmer, and files with the extension **.flash* are used for the web firmware upgrade.

Note: Features defined in *$UM/systems/$PLATFORM/include/configs/config_loader.h* and *$UM/systems/$PLATFORM/include/configs/config_umweb.h* can be selectively added or removed before building the firmware. Check the string “Features defined below can be selectively add or removed before building the image” in *config_loader.h* and *config_umweb.h*. *$PLATFORM* can be bcm95333x, bcm95340x, bcm95343x, or bcm95346x.

Build the Image File for UM+

BCM5333X Platform

1. Go to the *$UM/systems/bcm95333x* directory.
2. Type the following to build the firmware (*bcm95333x-umplus.bin*).
   ```
   make clean; make target=umplus
   ```
3. Use flash programmer to program *bcm95333x-umplus.bin* into the flash. Currently, the BCM953334K and BCM953394K use Micro N25Q256A13 serial flash.

BCM5340X Platform

1. Go to the *$UM/systems/bcm95340x* directory.
2. Type the following to build the firmware (*bcm95340x-umplus.bin*).
   ```
   make clean; make target=umplus
   ```
3. Use flash programmer to program *bcm95340x-umplus.bin* into the flash. Currently, the BCM953406K and BCM953456K use Micro N25Q512A83 serial flash.

BCM5343X Platform

1. Go to the *$UM/systems/bcm95343x* directory.
2. Type the following to build the firmware (*bcm95343x-umplus.bin*).
   ```
   make clean; make target=umplus
   ```
3. Use flash programmer to program *bcm95343x-umplus.bin* into the flash. Currently, the BCM953434K uses Winbond W25Q64FV serial flash.

BCM5346X Platform

1. Go to the *$UM/systems/bcm95346x* directory.
2. Type the following to build the firmware (*bcm95346x-umplus.bin*).
   ```
   make clean; make target=umplus
   ```
3. Use flash programmer to program *bcm95346x-umplus.bin* into the flash. Currently, the BCM956270K uses Macronix MX25L12835FMI-10G serial flash.

Note: Features defined in *$UM/systems/$PLATFORM/include/configs/config_umplus.h* can be selectively added or removed before building the firmware. Check the string “Features defined below can be selectively add or removed before building the image” in *config_umplus.h*. *$PLATFORM* can be bcm95333x, bcm95340x, bcm95343x, or bcm95346x.

Build the Image File for UM-Dumb

BCM5333X Platform

1. Go to the *$UM/systems/bcm95333x* directory.
2. Type the following to build the firmware (*bcm95333x-umdumb.bin*).
   ```
   make clean; make target=umdumb
   ```
3. Use flash programmer to program *bcm95333x-umdumb.bin* into the flash. Currently, the BCM953334K and BCM953394K use Micro N25Q256A13 serial flash.

BCM5340X Platform

1. Go to the *$UM/systems/bcm95340x* directory.
2. Type the following to build the firmware (*bcm95340x-umdumb.bin*).
   ```
   make clean; make target=umdumb
   ```
3. Use flash programmer to program *bcm95340x-umdumb.bin* into the flash. Currently, the BCM953406K and BCM953456K use Micro N25Q512A83 serial flash.

BCM5343X Platform

1. Go to the *$UM/systems/bcm95343x* directory.
2. Type the following to build the firmware (*bcm95343x-umdumb.bin*).
   ```
   make clean; make target=umdumb
   ```
3. Use flash programmer to program *bcm95343x-umdumb.bin* into the flash. Currently, the BCM953434K uses Winbond W25Q64FV serial flash.

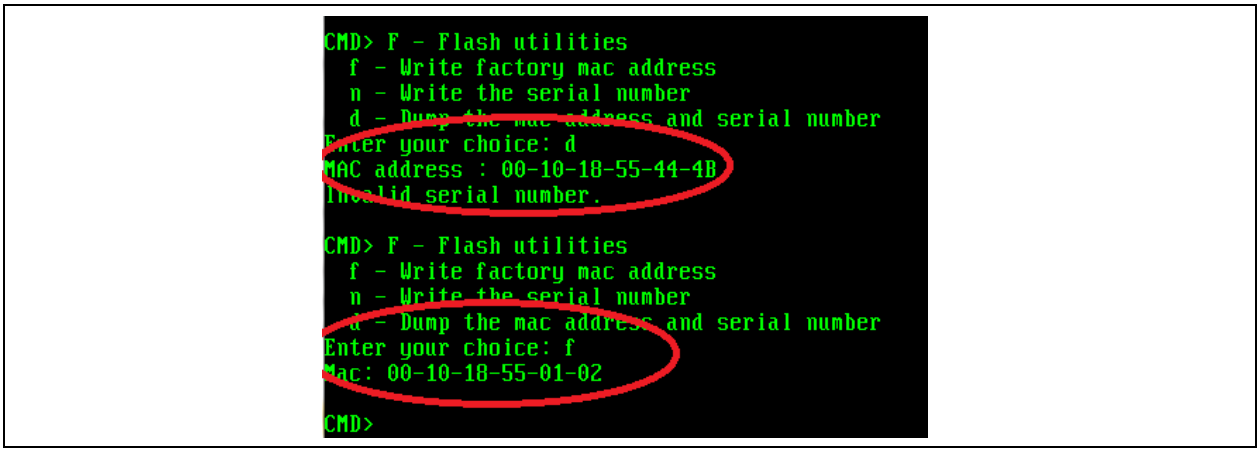
BCM5346X Platform

1. Go to the *$UM/systems/bcm95346x* directory.
2. Type the following to build the firmware (*bcm95346x-umdumb.bin*).
   ```
   make clean; make target=umdumb
   ```
3. Use flash programmer to program *bcm95346x-umdumb.bin* into the flash. Currently, the BCM956270K uses Macronix MX25L12835FMI-10G serial flash.

Note: Features defined in *$UM/systems/$PLATFORM/include/configs/config_umdumb.h* can be selectively added or removed before building the firmware. Check the string “Features defined below can be selectively add or removed before building the image” in *config_umdumb.h. $PLATFORM* can be bcm95333x, bcm95340x, bcm95343x, or bcm95346x.

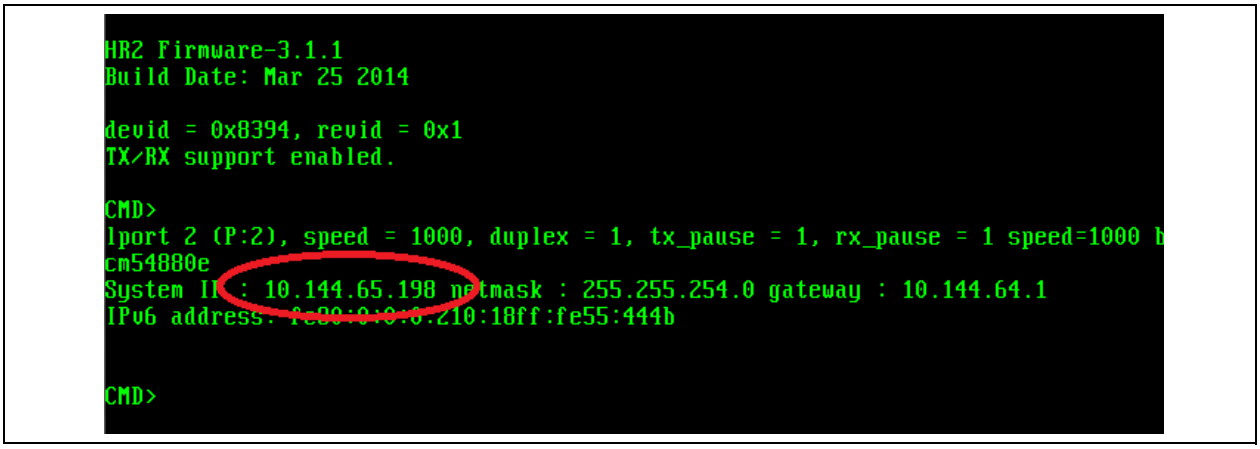
Device Bootup

1. Use `flash programmer` to program the image (*bcm95333x.image, bcm95340x.image, bcm95343x.image* or *bcm95346x.image*) into the flash.
2. After the device boots-up, change the device's MAC address using CLI commands. Type `F->f`.
3. Enter the new MAC address (e.g., `00-10-18-55-01-02`), then reboot the device to allow the new MAC address to take effect.
4. Type `F->d` to ensure that the MAC address has changed.



Web Interface

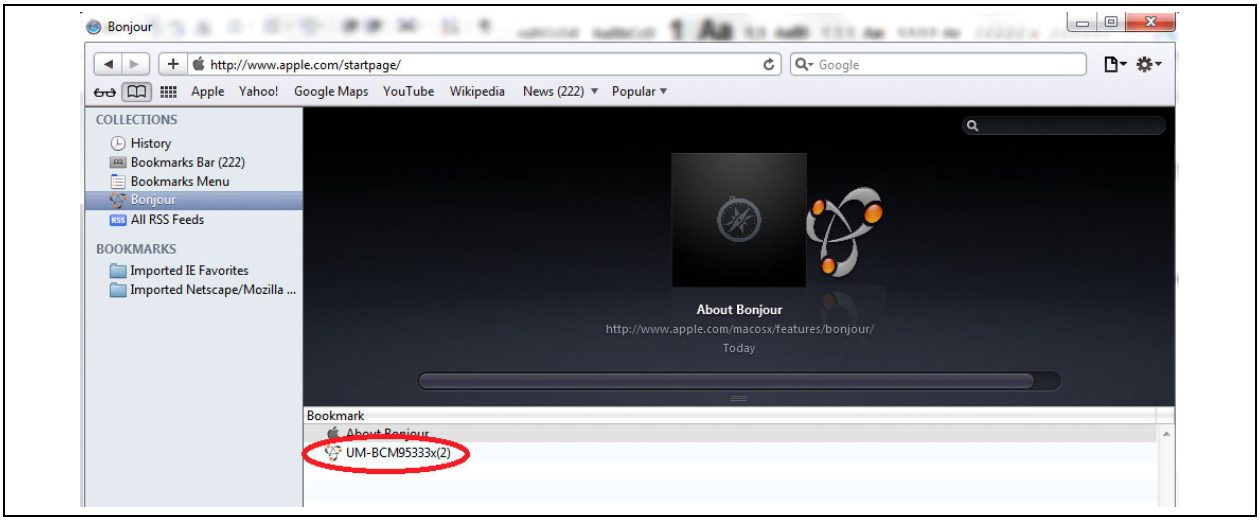
1. Type the URL with the IP address of the device in the web browser to enter the web interface. Then log in by typing `password` as the password. If you connect the device via a console (default baud rate is 9600 bps), the device's IP address will be displayed as shown in the following figure.



2. If you do not know the device's IP address or the device does not have a console, you can use the web browser with Bonjour support to discover the device's home page.

For Safari Web Browser

1. Open Safari.
2. Click **Show all bookmarks**.
3. Select Bonjour. You will see the device's HTTP service instance name under **Bookmarks**.
4. Click the bookmark, and Safari will connect to the device's home page.

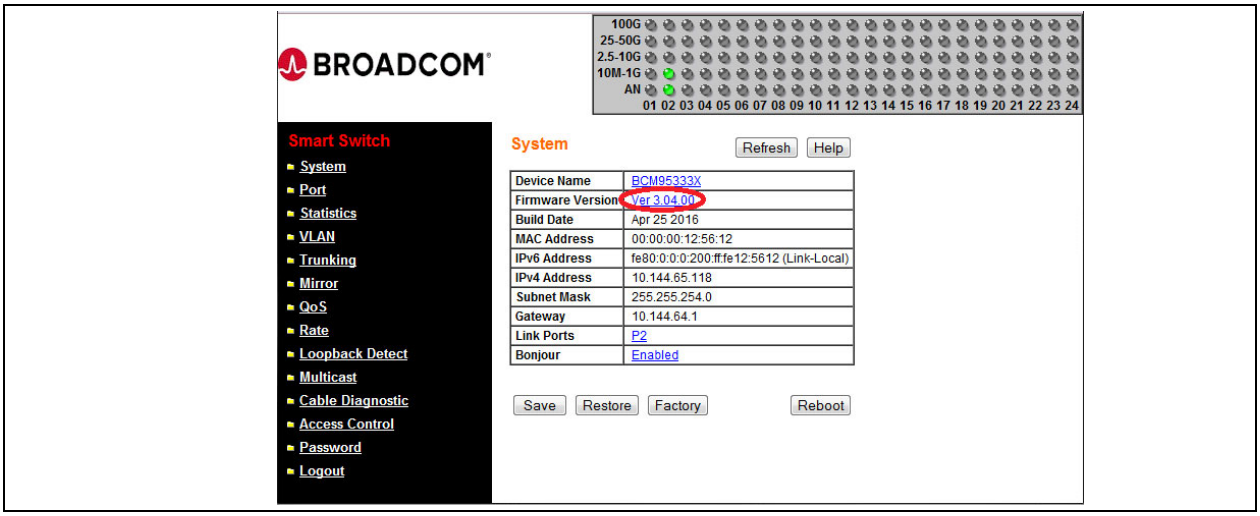


For Internet Explorer Web Browser

1. Install Bonjour for Windows (http://download.cnet.com/Bonjour-for- Windows/3000-18507_4-93550.html) to support Bonjour on Microsoft's Internet Explorer (IE) web browser.
2. Open the IE web browser and enable Bonjour from **Manage Add-ons**.
3. Click the device's HTTP service name from the left frame of Bonjour.

Firmware Upgrade

1. Use the web browser to connect to the device's home page.
2. Click the hyperlink from **Firmware Version**.



3. Click **OK** when it asks to confirm the firmware operation.
4. Wait until the button becomes **Continue**, and then click it.
5. If you do not want to upgrade the firmware, click **Abort** (and skip the remaining steps).
6. To upgrade the firmware, choose the firmware image file (*bcm95333x-umweb.flash, bcm95340x-umweb.flash, bcm95343x-umweb.flash* or *bcm95346x-umweb.flash*) and click **Upgrade**.
7. Wait until **Firmware Upgrade Completed** appears, then click **Continue**.

Note: Do not close the browser or power off the device before the upgrade is complete. Otherwise, the firmware image or serial flash could become corrupted.

Note: The IP address might be changed after reboot. See "Web Interface" on page 26 for information on the IP address.

Generate Code for Web Pages

1. Prepare the HTML file, for example, *$UM/src/appl/web/html/left.htm*.
2. Go to the directory *um/tools/web* and copy file *left.html* to this folder.
3. Type the following to generate files *left_htm.c*, *left_htm.h,* and *sspmacro_feature.h* based on the HTML file.
   `perl sspgen.pl left.htm`
4. Manually copy generated files *left_htm.c* and *left_htm.h* to *$UM/src/appl/web/content*.
5. Manually add new defines SSPMACRO_FEATURE_XXX in the generated file *sspmacro_feature.h* into same file under directory *$UM/src/appl/web/content*.

Generate Code for Xcommands

1. Define command syntax and describe the syntax in the XML file. For example, *$UM/src/appl/xcommands/switch/config.xml*.
2. Go to the directory *$UM/systems/$PLATFORM*. *$PLATFORM* could be *bcm95333x, bcm95340x, bcm95343x* or *bcm95346x*.
3. Type the following to generate files *xccxt_global_builders.c*, *xccxt_global_enums.h*, *xccxt_global_handlers.c* and *xccxt_table_global.c* under directory *$UM/src/appl/xcommands/generated* based on the XML file:
   `make xcommands`
4. Manually copy the functions or paths in the generated files *xccxt_global_builders.c* and *xccxt_global_handlers.c* to the same files under callback directory *$UM/src/appl/xcommands/callback* and revise it.

Cisco-Like CLI Interface

A user can display a Cisco-like CLI shell by typing `x` under the prompt `CMD>`. The Cisco-like CLI provides two accounts: admin (password is “password”) and guest (no password). The admin account can execute all levels of commands. The guest account can execute guest-level commands only, which includes the show commands in our example. UM software provides a few examples of Cisco-like CLI commands for reference.

Port Configuration

BCM5333X Platform

Figure 1 shows the front panel port numbers for the BCM953334K board.

Ports 1–24 are 1G copper mode with auto-negotiation.

Figure 1: Board BCM953334K Front Panel Port Number

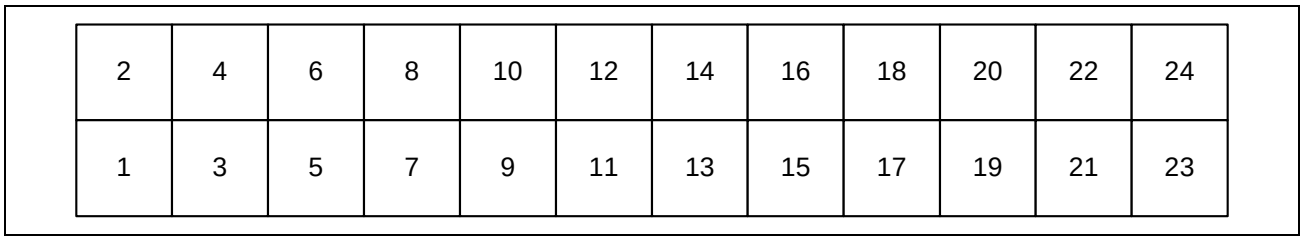
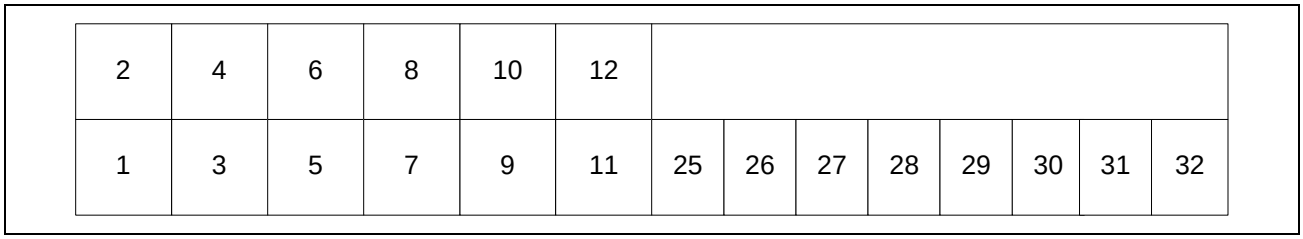


Figure 2 shows the front panel port numbers for the BCM953394K board.

Ports 1, 3, 5, 7, 9, and 11 are 1G copper mode with auto-negotiation. Ports 25–28 are 10G fiber mode with forced speed. Ports 29–32 are 1G fiber mode with auto-negotiation, clause 37. Ports 2, 4, 6, 8, 10, and 12 are unused ports.

Figure 2: Board BCM953394K Front Panel Port Number



BCM5340X Platform

Figure 3 shows the front panel port numbers for the BCM953406K board.

Ports 1–12 are 1G fiber mode with auto-negotiation, clause 37. Ports 13–24 are 10G fiber mode with forced speed.

Figure 3: Board BCM953406K Front Panel Port Number

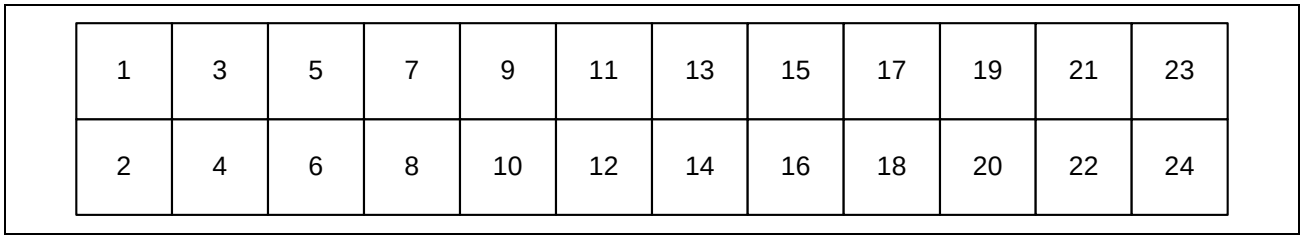
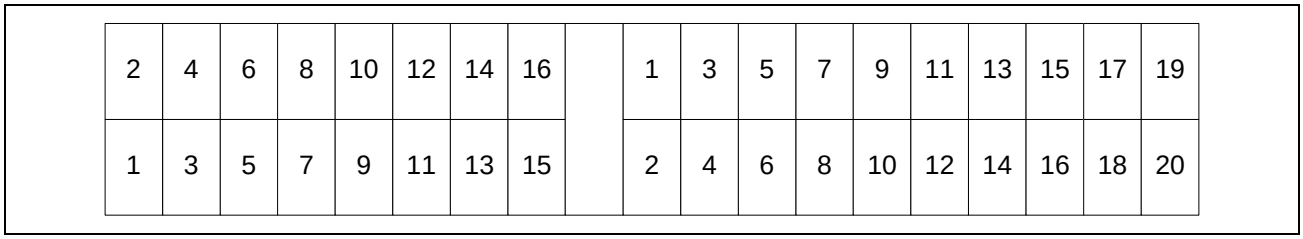


Figure 4 shows the front panel port numbers for the BCM953456K board.

The left 16 ports are 1G copper mode with auto-negotiation. The right ports 1–8 are 1G fiber mode with auto-negotiation, clause 37. Right ports 9, 11, 13, and 15 are 10G fiber mode with forced speed. Right ports 10, 12, 14, 16, and 17–20 are unused ports.

Figure 4: Board BCM953456K Front Panel Port Number



BCM5343X Platform

Figure 5 shows the front panel port numbers for the BCM953434K board.

If QTC ports are programmed in QSGMII mode (default mode):

- Ports 1–24 are 1G copper mode with auto-negotiation.
- Right ports R1 and R2 are unused.

If QTC ports are programmed in SGMII mode:

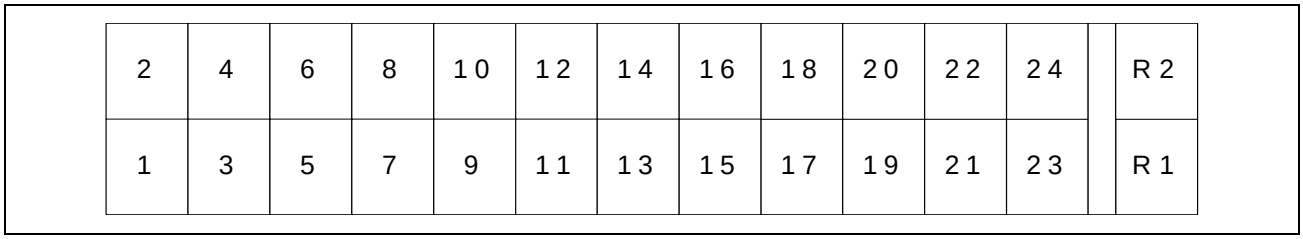
- Ports 1–16 are 1G copper mode with auto-negotiation.
- Ports 17–24 are unused.
- Right ports R1 and R2 are SGMII mode with auto-negotiation. In this version, only 1G speed is supported.

Note: Close pins 1 and 2 of J3402 and pins 3 and 4 of J3402 to let QTC ports work in SGMII mode for hardware.

Note: Use configuration "*qtc_interface=0x2*" to let QTC ports work in SGMII mode for software. Refer to "Vendor Configuration" on page 36 for detailed procedures.

Note: Fiber mode on QTC ports is supported from UM-3.3.2.

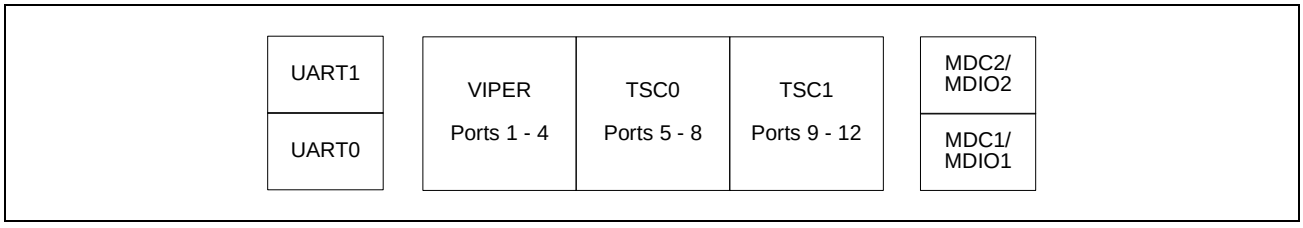
Figure 5: Board BCM953434K Front Panel Port Number



BCM5346X Platform

Figure 6 shows the front panel port numbers for the BCM956270K board with BCM53460 SKU. Ports 1–4 are 1G fiber mode with auto-negotiation, clause 37. Ports 5–12 are 10G fiber mode with forced speed.

Figure 6: Board BCM956270K Front Panel Port Number



Dual Image

Dual image is enabled by default. To build the loader and firmware without dual image ability, remove CFG_DUAL_IMAGE_INCLUDED in *config_loader.h* and *config_umweb.h*.

Table 3 lists the start address of firmware 1 and firmware 2 partitions. See “Memory Layout ” on page 15 for details.

Table 3: Start Address of Firmware 1 and Firmware 2 Partitions

Flash Partition	*BCM5333X Platform*	*BCM5340X/BCM5343X Platform*	*BCM5346X Platform*
Firmware 1	0X1C06-0000	0xF010-0000	0X1C10-0000
Firmware 2	0X1C13-0000	0xF028-0000	0X1C28-0000

The following is a simple test of enabling dual image through a firmware upgrade on the BCM5343x platform. The red information is related to dual image.

1. First, boot after using the flash programmer to program the um-3.3.0 image:

```
HURRICANE3 loader-3.3.0
Flash detected: W25Q64CV
Flash image is version 3.3.0 for board BCM95343X
Flash image is 575688 bytes, chksum 7DB7, version 3.3.0 for board BCM95343X
Load program at 0xF0100040..

HURRICANE3 umweb-3.3.0
Flash detected: W25Q64CV

devid = 0x8434, revid = 0x1
TX/RX support enabled.
Some of current saved settings are invalid. Loading factory defaults.....

CMD> D - Show dual image info
Image   Version   Active
------------------------
  1     3.3.0       Y
  2     NA          -
```

2. During the first firmware upgrade via the UM-Web:

```
HURRICANE3 loader-3.3.0
Flash detected: W25Q64CV

devid = 0x8434, revid = 0x1
TX/RX support enabled.
System IP : 10.144.65.187 netmask : 255.255.254.0 gateway : 10.144.64.1

CMD>
lport 2 (P:10), speed = 10, duplex = 0, tx_pause = 0, rx_pause = 0, an = 1
IPv6 address: fe80:0:0:0:210:18ff:fe55:444b
Upgrading firmware at partition 2 address 0xf0280000
Flash image is version 3.3.0 for board BCM95343X
```

3. After the first firmware upgrade via UM-Web again:

```
CMD>
HURRICANE3 loader-3.3.0
Flash detected: W25Q64CV
Flash image is version 3.3.0 for board BCM95343X
Flash image is 575688 bytes, chksum 7DB7, version 3.3.0 for board BCM95343X
Flash image is version 3.3.0 for board BCM95343X
Flash image is 575688 bytes, chksum 7DB7, version 3.3.0 for board BCM95343X
Load program at 0xF0280040...

HURRICANE3 umweb-3.3.0
Flash detected: W25Q64CV

devid = 0x8434, revid = 0x1
TX/RX support enabled.

CMD> D - Show dual image info
Image   Version   Active
------------------------
  1     3.3.0       N
  2     3.3.0       Y
```

4. During the second firmware upgrade:

```
HURRICANE3 loader-3.3.0
Flash detected: W25Q64CV

devid = 0x8434, revid = 0x1
TX/RX support enabled.
System IP : 10.144.65.187 netmask : 255.255.254.0 gateway : 10.144.64.1

CMD>
lport 2 (P:10), speed = 10, duplex = 0, tx_pause = 0, rx_pause = 0, an = 1
IPv6 address: fe80:0:0:0:210:18ff:fe55:444b
Upgrading firmware at partition 1 address 0xf0100000
Flash image is version 3.3.0 for board BCM95343X
```

5. After the second firmware upgrade:

```
HURRICANE3 loader-3.3.0
Flash detected: W25Q64CV
Flash image is version 3.3.0 for board BCM95343X
Flash image is 575688 bytes, chksum 7DB7, version 3.3.0 for board BCM95343X
Flash image is version 3.3.0 for board BCM95343X
Flash image is 575688 bytes, chksum 7DB7, version 3.3.0 for board BCM95343X
Load program at 0xF0100040...

HURRICANE3 umweb-3.3.0
Flash detected: W25Q64CV

devid = 0x8434, revid = 0x1
TX/RX support enabled.

CMD> D - Show dual image info
Image   Version   Active
-------------------------
  1     3.3.0       Y
  2     3.3.0       N
```

Minimum Storage Requirements

Table 4 lists the minimum storage requirements, based on the UM 3.4.0 release.

Table 4: UM 3.4.0 Minimum Storage Requirements

	UM+ (*.bin)		***UM-Web (*.image)***	
	BCM5333X	***BCM5340X/ BCM5343X/ BCM5346X***	***BCM5333X***	***BCM5340X/ BCM5343X/ BCM5346X***
Minimum flash size (Includes one firmware only)	512 KB	1 MB	–	–
Minimum flash size (Includes one loader and one firmware)	–	–	1 MB	2 MB
Minimum flash size with dual image (Includes one loader and two firmware)	–	–	2 MB	4 MB

Vendor Configuration 开发者配置

The Unmanaged Software provides a tool to insert the configuration into the precompiled firmware image (*.image) as well as a set of APIs to retrieve the configuration at runtime. The firmware image may perform different settings in the initialization based on the configuration.

Configuring the APIs

`$UM/include/$(CPU)/sal_config.h` reveals the APIs to get the value of certain configuration items which can be regarded as a specific type, such as uint8, uint16, and port bitmap.

The caller should give the name of the configuration item and use the proper function to get the configure value. If the given configuration item is not found at configuration space or an error occurs when getting the configuration, the APIs will return an error and not affect the content of the output pointer.

Function prototypes for the firmware APIs are shown below.

```
sys_error_t sal_config_pbmp_get(const char *name, pbmp_t *p)
int sal_config_bytes_get(const char*name, uint8* buf, int len)
sys_error_t sal_config_uint8_get(const char*name, uint8* byte)
sys_error_t sal_config_uint16_get(const char*name, uint16* hword)
sys_error_t sal_config_uint32_get(const char*name, uint32* word)
```

Using the Configuration Insert Tool

`$UM/tool/um_config_insert.pl` is the configuration insert tool. When given the firmware image and the configuration file, this tool will translate this configuration file into binary and insert it into the firmware image. An example of the tool is shown below.

```
$UM/tool/um_config_insert.pl -i bcm95340x.image -c config.um -force
```

`bcm95340x.image` is the firmware image. `config.um` is the configuration file. The option, `-force`, will insert the configuration regardless of the existence of the previous inserted configuration. For more information, execute `$UM/tool/um_config_insert.pl -h` to see the usage of the insert tool.

Usage: `um_config_insert.pl -image <image_file> [options]`:

- -image, `-i`: This specifies the image file where the configuration will be inserted.
- -config, `-c`: This optionally specifies the configuration file name. The default name is `config.um`.
- -force: This forces the previous configuration in the image file to be overwritten.
- -verbose: This shows the debug log.
- -h: This shows the configuration file usage.

A sanity check is performed at the beginning of tool execution to ensure that the configuration insert procedure is correct. The sanity check includes:

1. Checking the configuration file syntax.
2. Verifying the availability of the firmware image space for configuration storage.
3. Checking if `board_name` is defined in configuration file. The `board_name` in the configuration space should be equal to the `CFG_BOARDNAME` defined in Makefile.

$UM/tools/config.um

```
#
# Vendor Configuration
#
# Each entry in the file consists of a single line of the form:
#    <Parameter>=<Value>
# UM software provides a tool (um_config_insert.pl) to insert the Vendor
# configuration (config.um) into the precompiled firmware image (*.image).
# Then the firmware image may do different settings in the intialization
# based on the configuration.
#
# Usage to insert Vendor Configuration into the precompiled firmware
#      um_config_insert.pl -image  <image_file> [options]
#          -image, -i: specify the image file where the config will be inserted
#          -config, -c: optionally specify config file name,
#                           default name is "config.um"
#          -force: force to overwrite previous configuration in the image file
#          -verbose: show more debug log
#          -h: show usage
#

#     Board Name : Board name checking will be enabled if board_name is set.
#                  For board name checking, it would check whether it is equal
#                  to the value of CFG_BOARDNAME defined in
#                  system/bcmxxxxx/Makefile.
#board_name=BCM95340X

#     SKU Option : Option of SKU, value is from 1 ~ X.
#                  Please get SKU option from Programmer's Register Reference
#                  Guide.
#sku_option=2

#     Serial LED
#         - led_option : Option of serial LED micro code, value is from 1 ~ 3.
#             1: Left LED : Link              Right LED : TX/RX activity
#             2: Left LED : TX/RX activity    Right LED : Link
#             3: Customer LED uCode
#         - led_program : Customer LED uCode
#led_option=3
#led_program=led.hex
```

```
#     Parallel LED Setting
#         - Overide the setting of register LED_CONTROL and LED_SELECTOR
#           for external PHY.
#phy_led1_mode=0xF
#phy_led2_mode=0xF
#phy_led3_mode=0xF
#phy_led4_mode=0xF
#phy_led_ctrl=0x8
#phy_led_select=0x0

#     Reset Button
#         - reset_button_enable :  0 (disable) / 1 (enable reset button feature)
#         - reset_button_gpio_bit : Set GPIO bit for reset button, value is
#                                   from 0 ~ 7.
#         - reset_button_polarity : 0 (active low) / 1 (active high)
#reset_button_enable=1
#reset_button_gpio_bit=4
#reset_button_polarity=1

#     Per Port Configuration : Value is a set of logical port number,
#                              for example 2-5,10-13.
#valid_logical_ports=2-25
#speed_1000_logical_ports=2-25
#speed_2500_logical_ports=2-25
#speed_5000_logical_ports=2-25
#speed_10000_logical_ports=2-25
#phy_an_logical_ports=2-25
#phy_cl73_logical_ports=2-25
#phy_cl37_logical_ports=2-25

#     QTC interface
#         - qtc_interface : Different modes for qtc interface, value is from 1 ~ 3.
#             1: QSGMII mode
#             2: SGMII mode
#             3: Fiber mode
#qtc_interface=1

#     TSCE interface
#         - tsce_interface : Different modes for tsce interface, value is 1 or 2.
#             1: SGMII mode
#             2: Fiber mode
#tsce_interface=1

#     VIPER interface
#         - viper_interface : Different modes for viper interface, value is 1 or 2.
#             1: SGMII mode
#             2: Fiber mode
#viper_interface=1

#     Config "static IP address" or dhcp
#         - ifconfig=IP_ADDR/NETMASK/GATEWAY or ifconfig=dhcp
#
#           For example1, ifconfig=192.168.0.239/255.255.255.0/192.168.0.254
#           For example2, ifconfig=dhcp, it is same as if ifconfig is not configured
#
#ifconfig=192.168.0.239/255.255.255.0/192.168.0.254
```

Note: For logical port number, refer to the “Port and PHY Configuration” section of document “Unmanaged Software Porting Guide” (Reference [1] on page 9) for detailed procedures.

Example 1

The following section provides an example to insert the configuration “*qtc_interface=2*” into the release image *bcm95343x.image*.

1. Go to the tools directory `$UM/tools` and copy the *bcm95343x.image* from the image directory.
2. Add “`qtc_interface=2`” to `config.um`.

   In `config.um`:

   ```
   # QTC interface
   # - qtc_interface : Different modes for qtc interface, value is from 1 ~3.
   # 1: QSGMII mode
   # 2: SGMII mode
   # 3: Fiber mode
   qtc_interface=2
   ```

3. Use the `um_config_insert.pl` tool to insert the configuration into the precompiled firmware image (*.image). The output file will be the same name *.image. It is “bcm95343x.image” in this example.

   ```
   um_config_insert.pl -i bcm95343x.image
   ========================================================================
   UM signature is found at bcm95343x.image
   Detail parameters of image are shown below
   Board name = BCM95343X
   Firmware Type = loader
   Version = 0x3030000
   Config base address = 0xfd000
   Config maximun size = 0x1000
   ========================================================================
   Perform config.um Parsing ...
   qtc_interface=>2
   ========================================================================
   Config inserted successfully
   ========================================================================
   ```

4. Use the flash programmer to program the output image “`bcm95343x.image`” into serial flash.
5. Use the `F->l` command to list the configuration after the system boots up. It is for debug only.

   ```
   CMD> F - Flash utilities
   f - Write factory mac address
   n - Write the serial number
   d - Dump the mac address and serial number
   s - Set nvram variable
   g - Get nvram variable
   l - list all nvram variable
   r - Remove nvram variable
   c - Commit nvram variable bindings
   Enter your choice: l
   qtc_interface=2
   ```

Example 2

The following section provides an example of how to use an LED uCode. In this case, two configurations (`led_option=3` and `led_program=led_example.hex`) must be inserted into the precompiled image bcm95343x.image.

1. Prepare the LED uCode (*.hex).
   a. Prepare an `.asm` file. It is `led_example.asm` in this case.
   b. Generate an assembler that can convert the `.asm` file into .hex files.
      ```
      make ledasm' under $UM/tools/led/tools
      ```
   c. Use assembler to covert the `.asm` file into .hex files. The `.asm` file needs to be copied to `$UM/tools/led/tools`.
      ```
      ledasm led_example' under $UM/tools/led/tools
      ```
2. Go to the tools directory `$UM/tools` and copy the `bcm95343x.image` from the image directory.
3. Copy the `.hex` file generated in Step 1 to `$UM/tools` also.
4. Add `led_option=3` and `led_program=led_example.hex` to config.um. In config.um:
   ```
   # Serial LED
   # - led_option : Option of serial LED micro code, value is from 1 ~ 3.
   # 1: Left LED : Link Right LED : TX/RX activity
   # 2: Left LED : TX/RX activity Right LED : Link
   # 3: Customer LED uCode
   # - led_program : Customer LED uCode
   led_option=3
   led_program=led_example.hex
   ```
5. Use the `um_config_insert.pl` tool to insert these two configurations into the precompiled firmware image (*.image). The output file will be the same name *.image. It is bcm95343x.image in this example.

   ```
   um_config_insert.pl -i bcm95343x.image
   ========================================================================
   UM signature is found at bcm95343x.image
   Detail parameters of image are shown below
   Board name = BCM95343X
   Firmware Type = loader
   Version = 0x3030000
   Config base address = 0xfd000
   Config maximun size = 0x1000
   ========================================================================
   ========================================================================
   Perform config.um Parsing ...
   led_option=>3
   led_program=>led_example.hex
   led_example.hex is loaded
   ========================================================================
   Config inserted successfully
   ========================================================================
   ```
6. Use the flash programmer to program the output image `bcm95343x.image` into serial flash.

7. Use the `F->l` command to list the configuration after the system boots up. It is for debug only.

```
CMD> F - Flash utilities
f - Write factory mac address
n - Write the serial number
d - Dump the mac address and serial number
s - Set nvram variable
g - Get nvram variable
l - list all nvram variable
r - Remove nvram variable
c - Commit nvram variable bindings
Enter your choice: l
led_option=3
led_program=led_example.hex
```

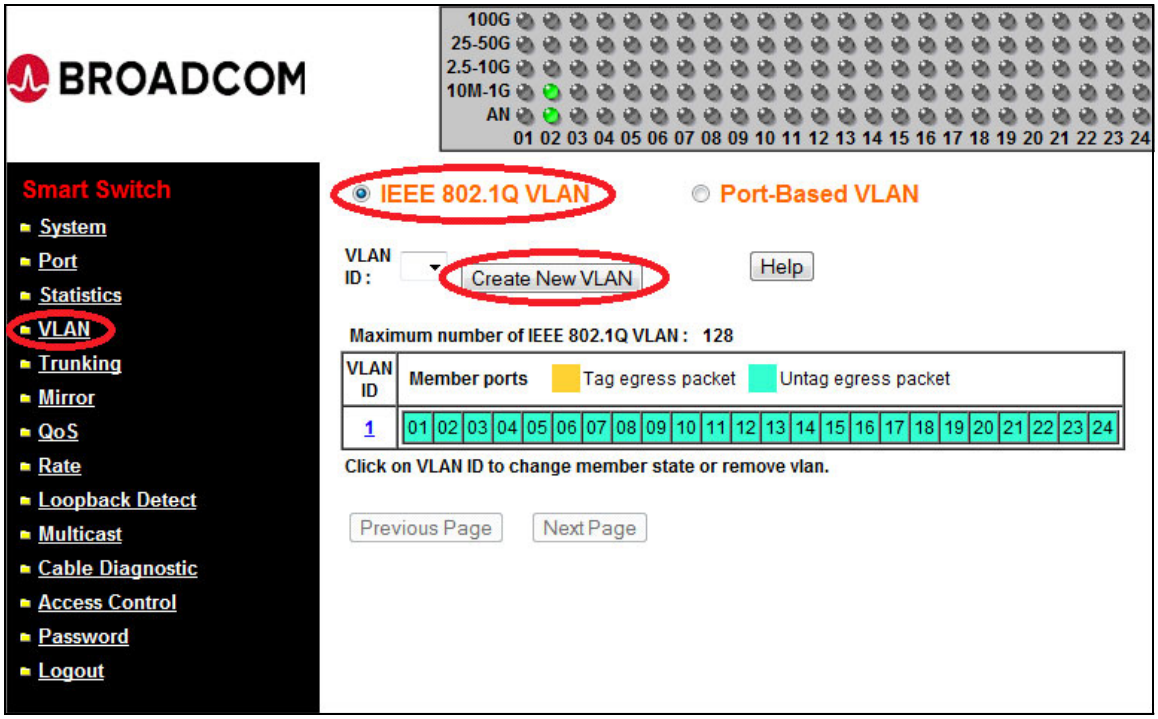
Configure VLANs by Web GUI

802.1Q VLAN

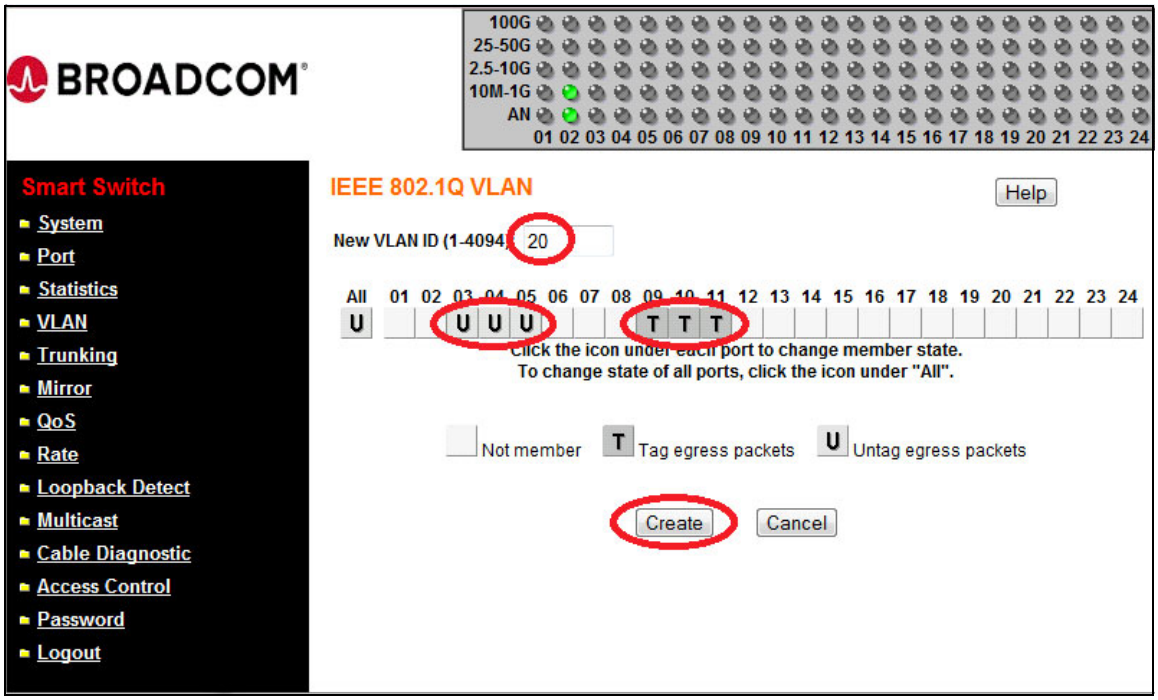
Click the VLAN hyperlink to display the VLAN web page. Choose **IEEE 802.1Q VLAN** to start setting the 802.1Q VLAN. The default setting is 802.1Q VLAN for UM-Web. By changing the VLAN type, the configuration related to the VLAN will be erased.

Creating a New 802.1Q VLAN

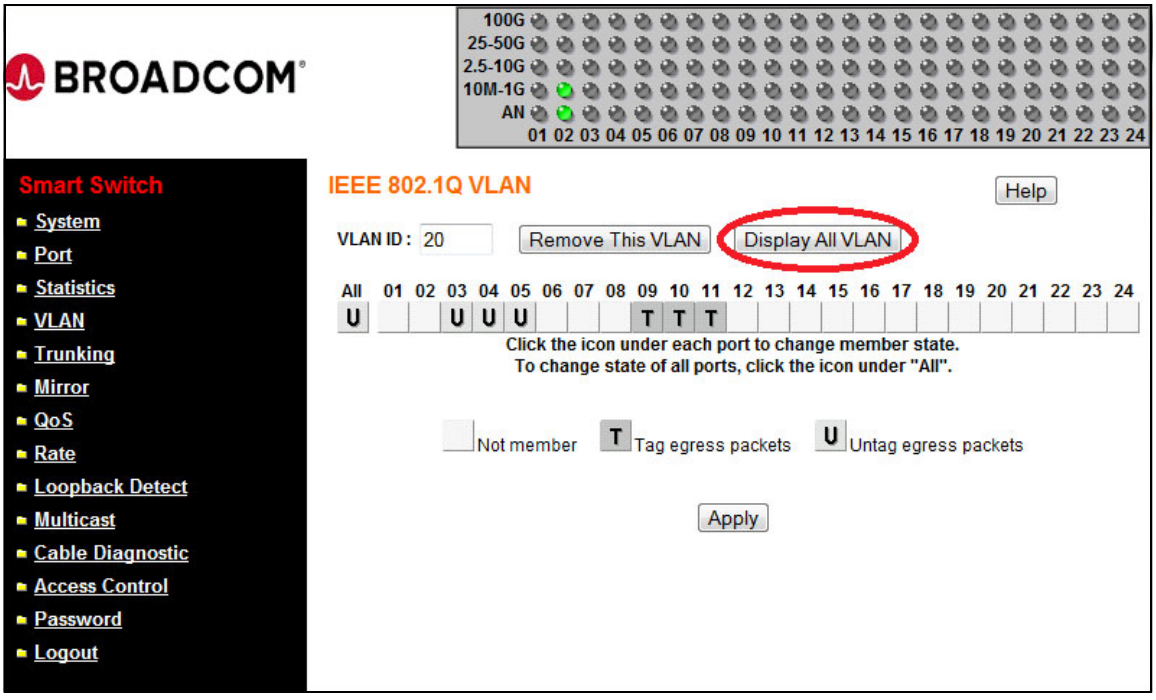
1. Click **Create New VLAN** to enter another page for creating a new VLAN.



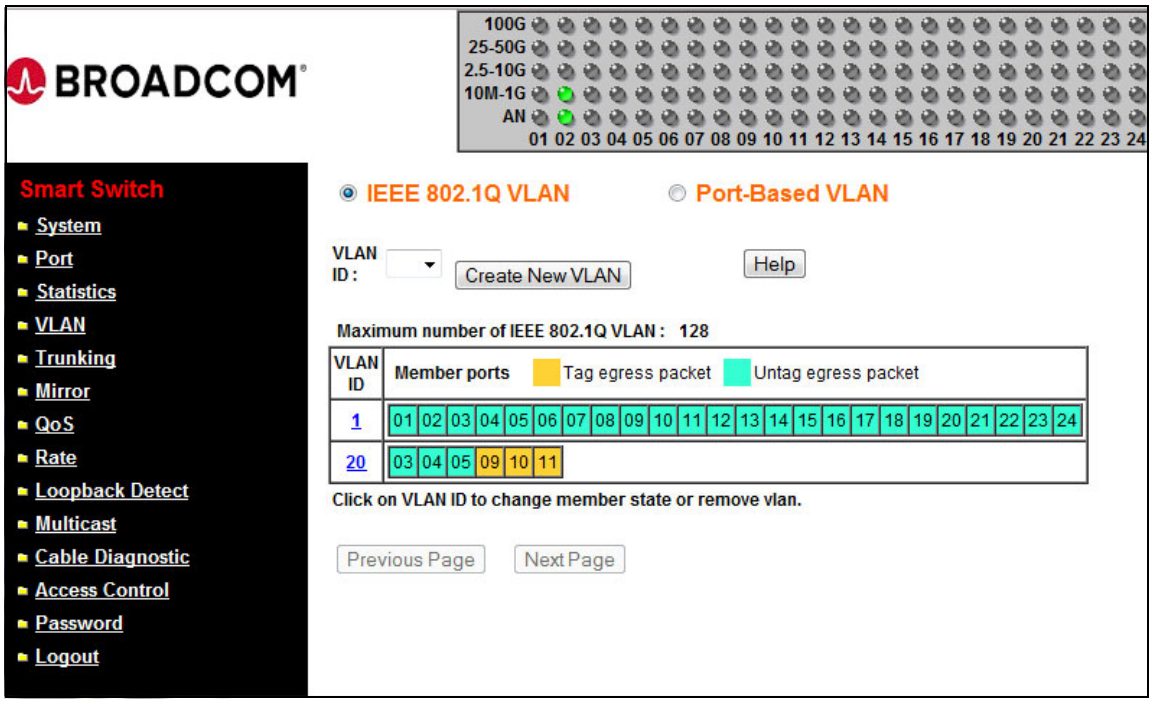
2. Type the new VLAN ID in **New VLAN ID** and select the ports belonging to this new VLAN with a tag or untag attribute. Then click **Create**.



3. After clicking Create, the page shows **Display ALL VLAN**. Click it to display all VLANs' information.



4. Click on **VLAN ID 20** to change the member ports of this VLAN or to remove this VLAN.

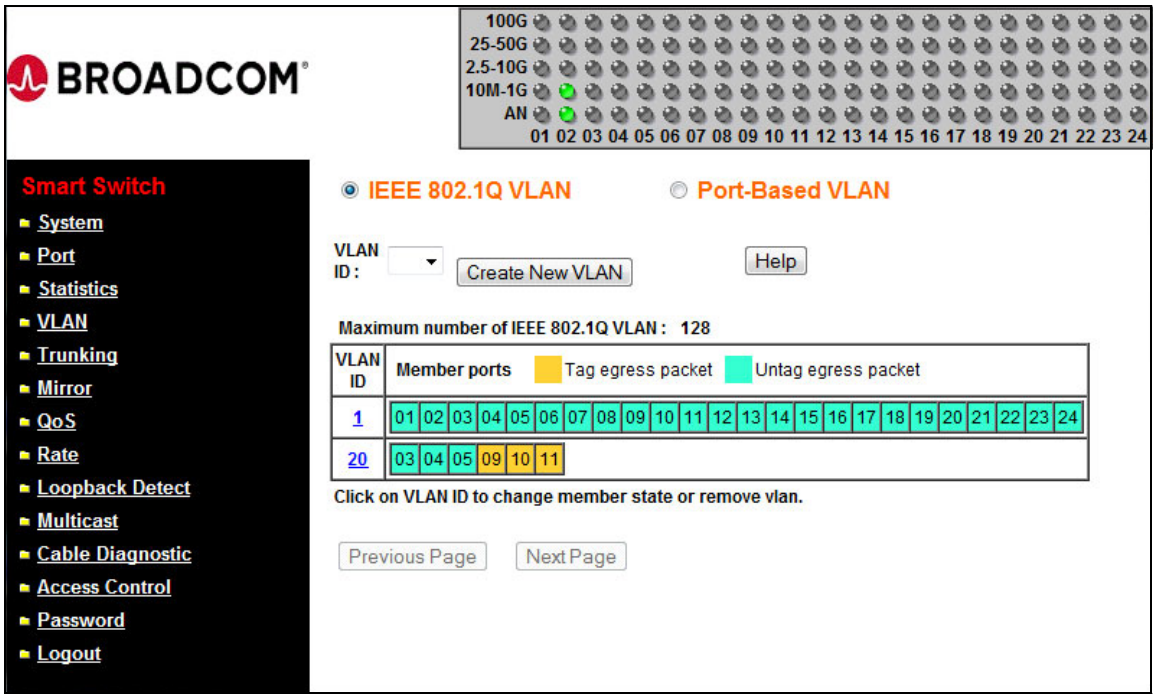


Modify or Delete an 802.1Q VLAN

To change member ports of a VLAN, select the port number to select the desired state: Not member, Tag member, or Untag member.

To delete a VLAN, select **VLAN_ID** and then click **Remove This VLAN**.

To modify a VLAN, select **VLAN_ID**, modify VLAN members, and then click **Apply**.

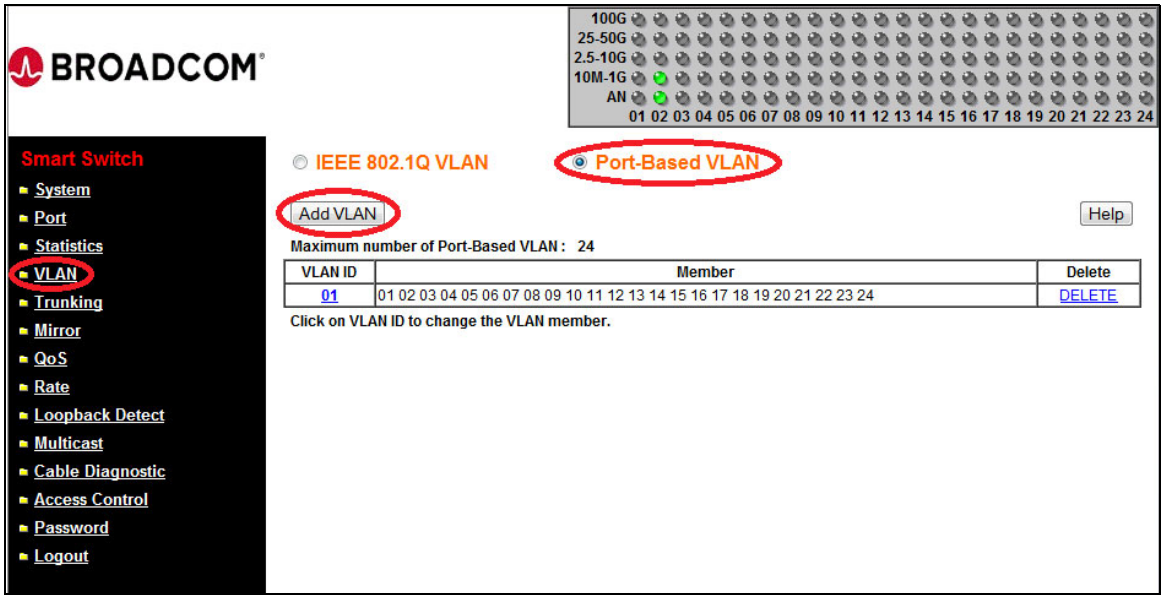


Port-Based VLAN

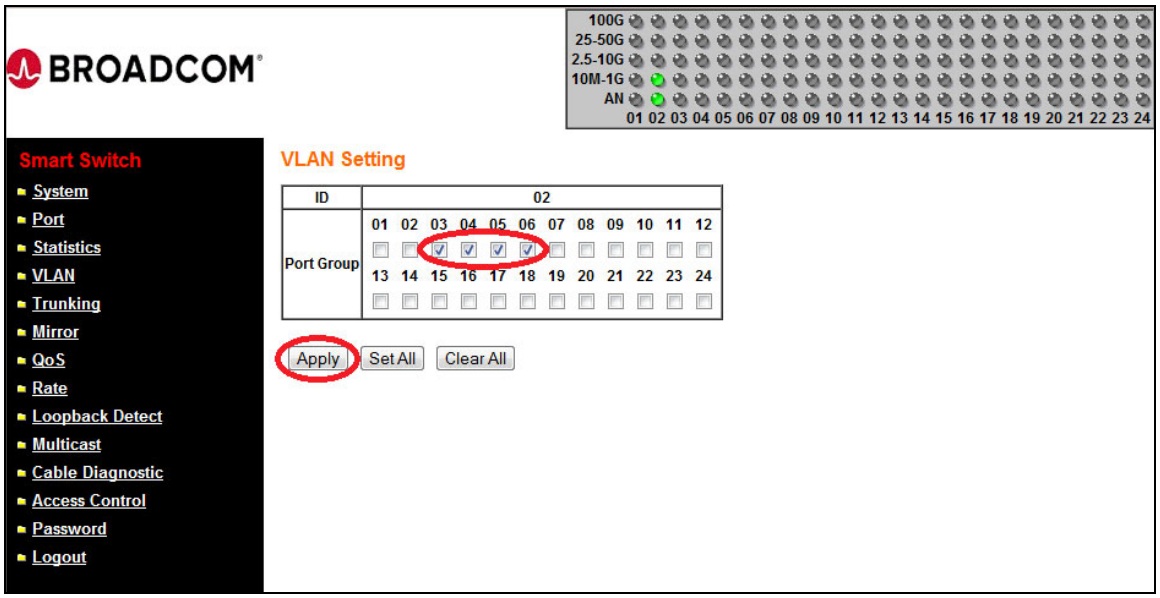
Click the VLAN hyperlink to display the VLAN web page. Choose **Port-Based VLAN** to start setting the Port-Based VLAN. By changing the VLAN type, the configuration related to the VLAN will be erased.

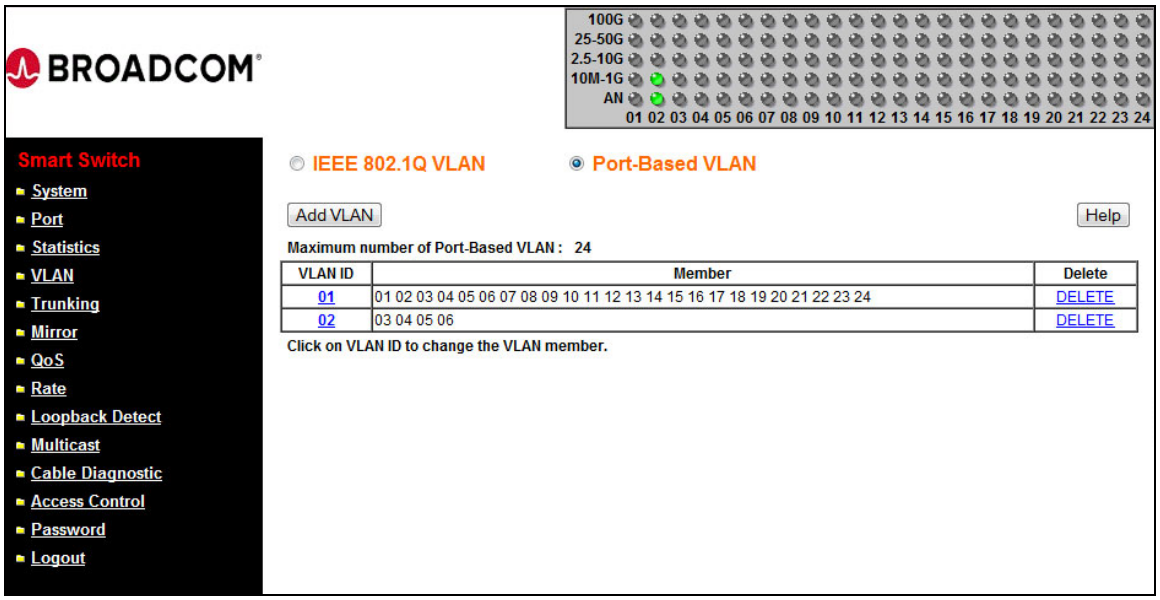
Create a New Port-Based VLAN

1. Click **Add VLAN** to enter another page for creating a new VLAN.



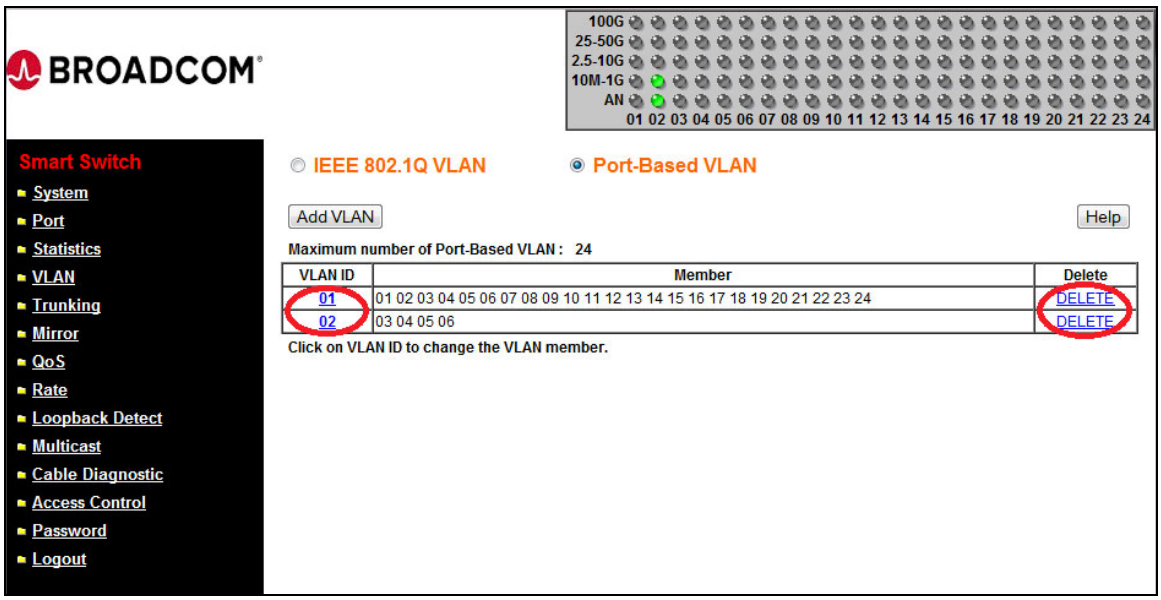
2. Select ports belonging to this new VLAN, then click **Apply**.





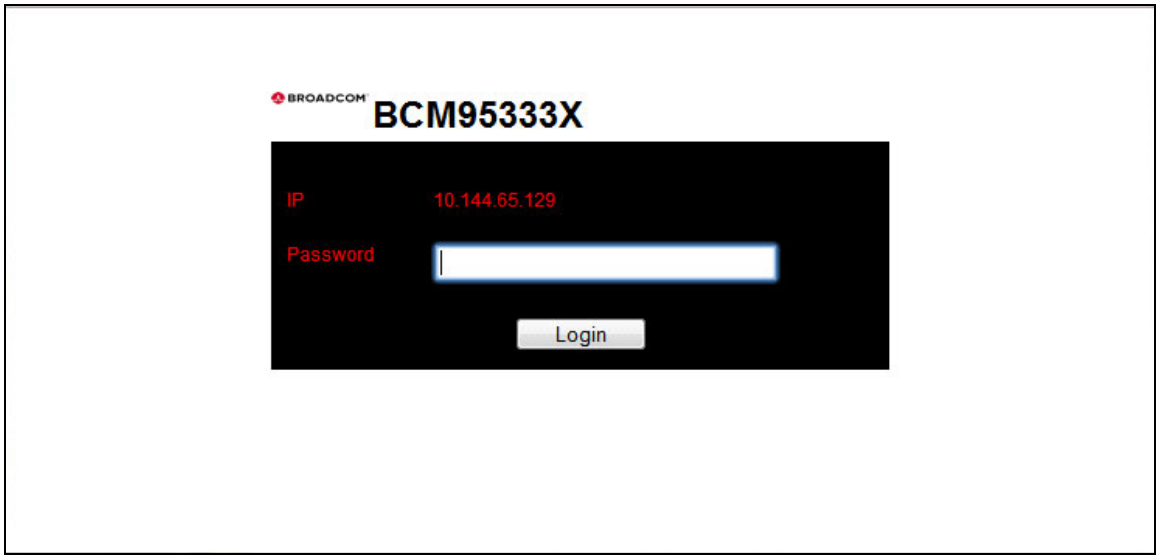
Modify or Delete a Port-Based VLAN

Click on **VLAN ID** to change member ports of this VLAN, or click **DELETE** to remove this VLAN.



Additional Web Pages

At the Login page, type your password.

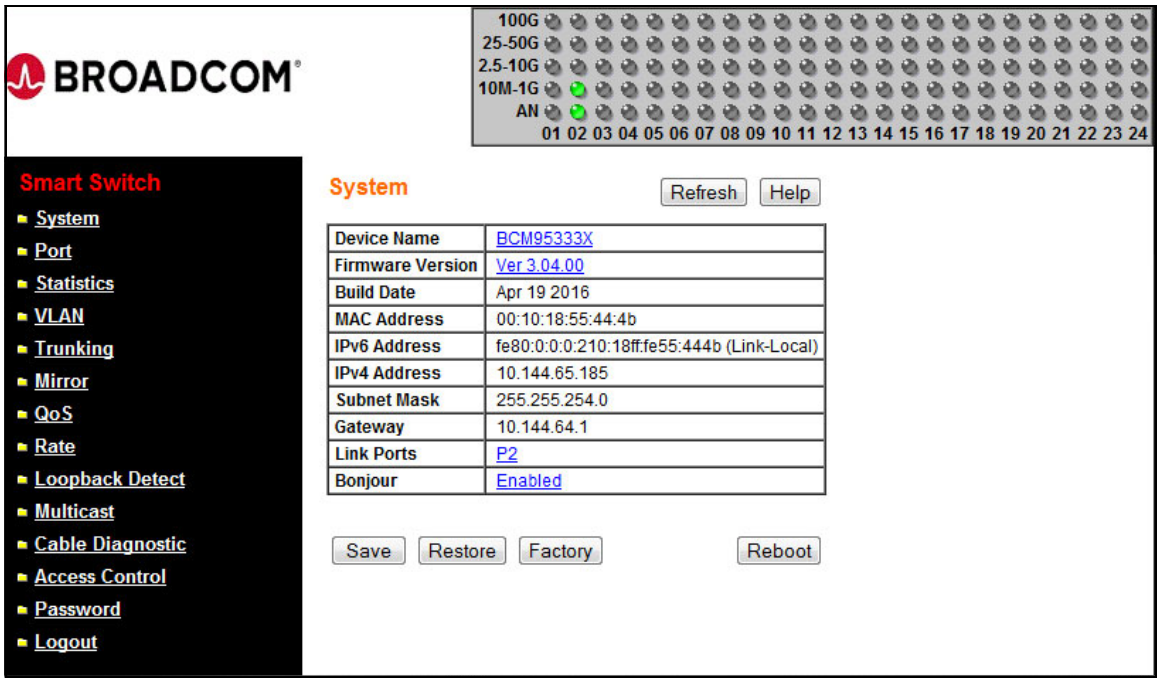


System Page

The system page is the first page after the login page.

Table 5: System Page Parameter Description

Parameter/Button	*Description*
Device Name	Click the device name to set a new device name.
Firmware Version	Revision ID of the system. Click firmware version to request a firmware upgrade. After confirming this request, the device will be rebooted to bootloader for upgrade process.
Build Date	Build date of this firmware.
MAC Address	MAC address of the in-band Ethernet interface.
IPv6 Address	IPv6 addresses of the system.
IPv4 Address	IPv4 address of the system.
Subnet Mask	Subnet mask of the IP address.
Gateway	Gateway address of the IP address.
Link Ports	Current link-up ports.
Bonjour	Click to change the current Bonjour mode.
Refresh	Retrieves all system information.
Save	Saves all configurations of the device to a file.
Restore	Restores all configurations of the device from a file.
Factory	Loads factory-default setting.
Reboot	Resets device.

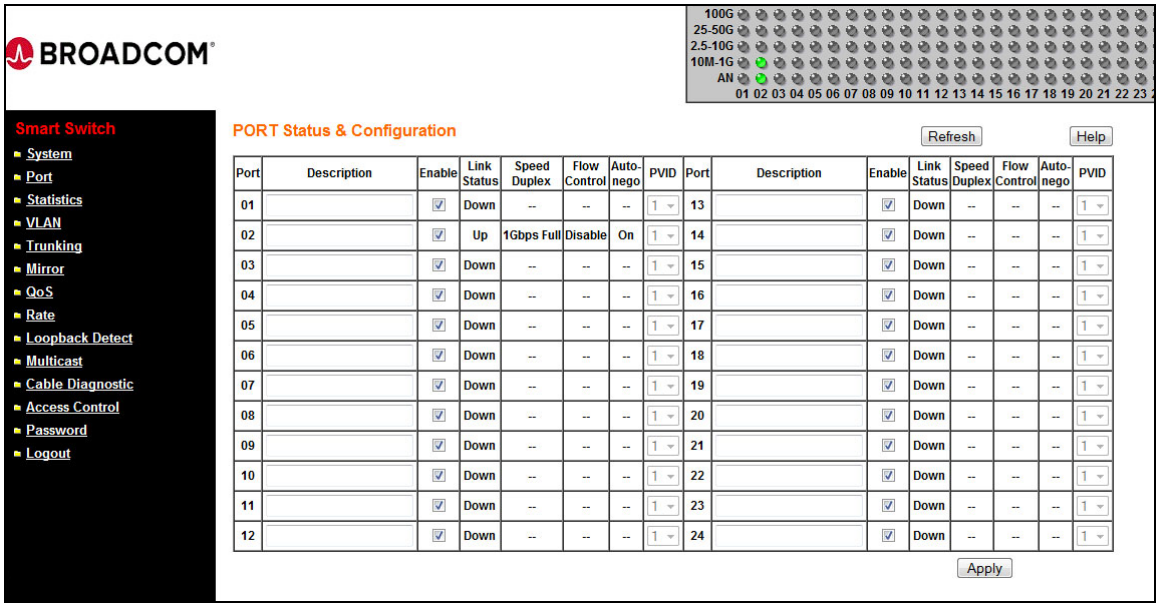


Port Status and Configuration Page

Port functions provide an overview of the system. The Port Status and Configuration page displays each port's status, such as link status, speed, duplex, flow control, and auto-negotiation.

Table 6: Port Status and Configuration Page Parameter Description

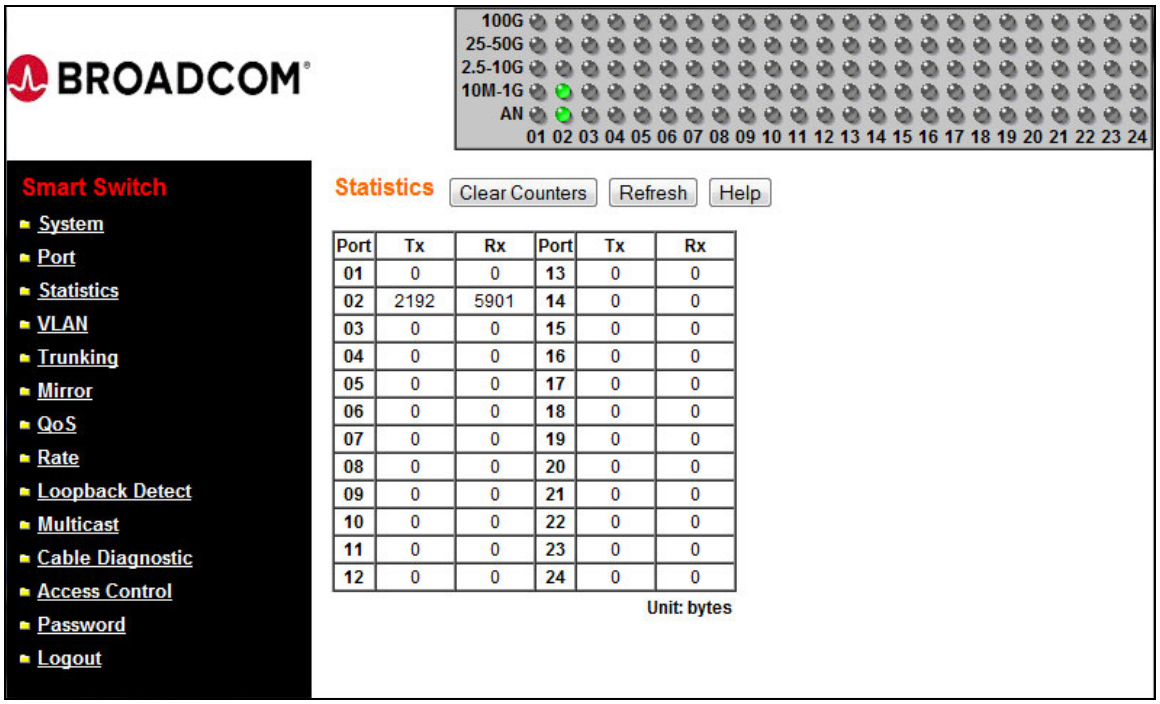
Parameter/Button	*Description*
Port Number	Indicates the port number.
Description	The description of the port.
Link Status	Displays the link status of the port.
Speed/Duplex	Indicates the duplex and speed of the port when it is linked. If the port link is down, there is no status display.
Flow Control	The flow control of the port when it is linked. If the port link is down, there is no status displayed.
Auto-nego	Displays the auto-negotiation status of the port.
PVID	Assigns a VLAN ID of the packet when packets arrive at the port for an untagged packet or a priority tagged packet.
Refresh	Retrieves port status and updates the parameters on the Port Status page.



Statistics Page

Table 7: Statistics Page Parameter Description

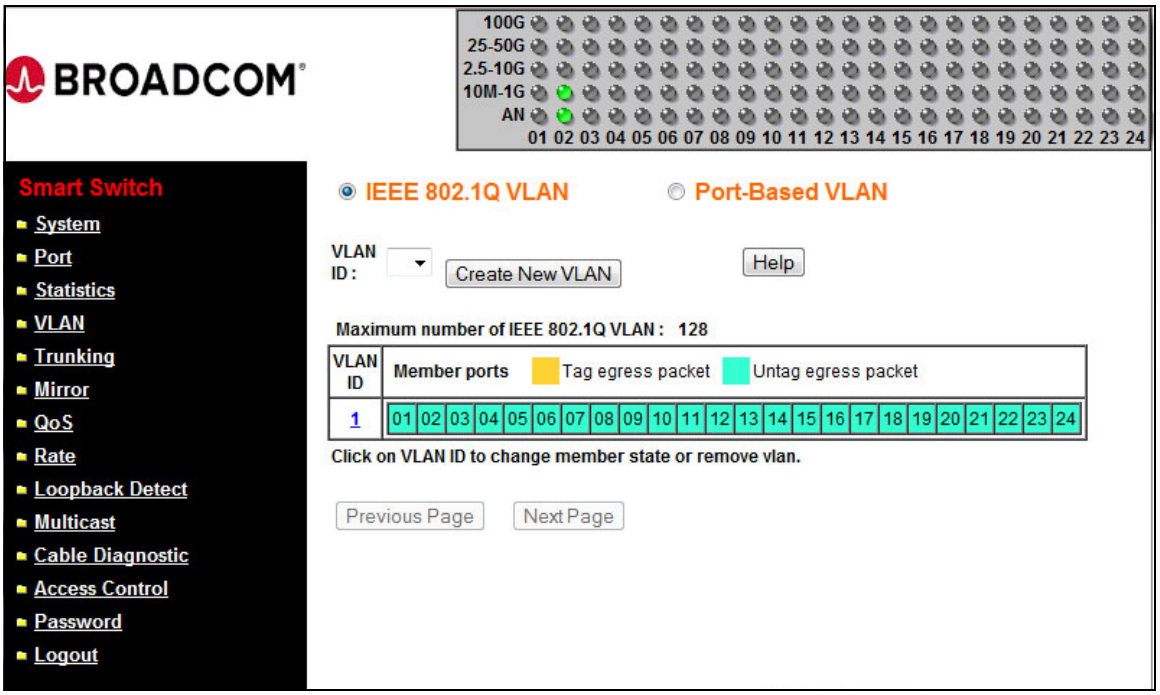
Parameter/Button	*Description*
Tx	Indicates the total packets transmitted at the port.
Rx	Indicates the total packets received at the port.
Clear Counters	Resets all counters to zero.
Refresh	Retrieves current count from the device and updates the Statistics page.



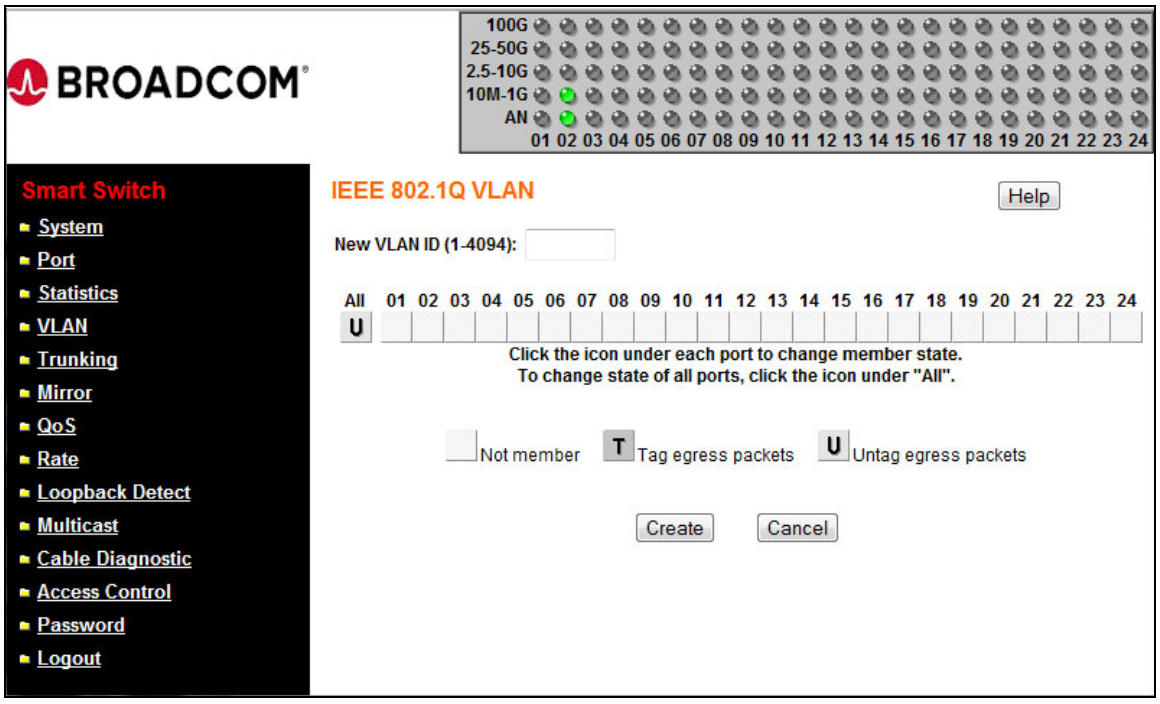
VLAN Page—IEEE 802.1Q VLAN

Table 8: VLAN Page — IEEE 802.1Q VLAN Parameter Description

Parameter/Button	*Description*
Member Ports	Member ports of the VLAN. There are two color symbols for each port: yellow for tag egress packet and aqua for untag egress packet.
Create New VLAN	Creates the VLAN.
VLAN ID	Changes member state or removes the VLAN.



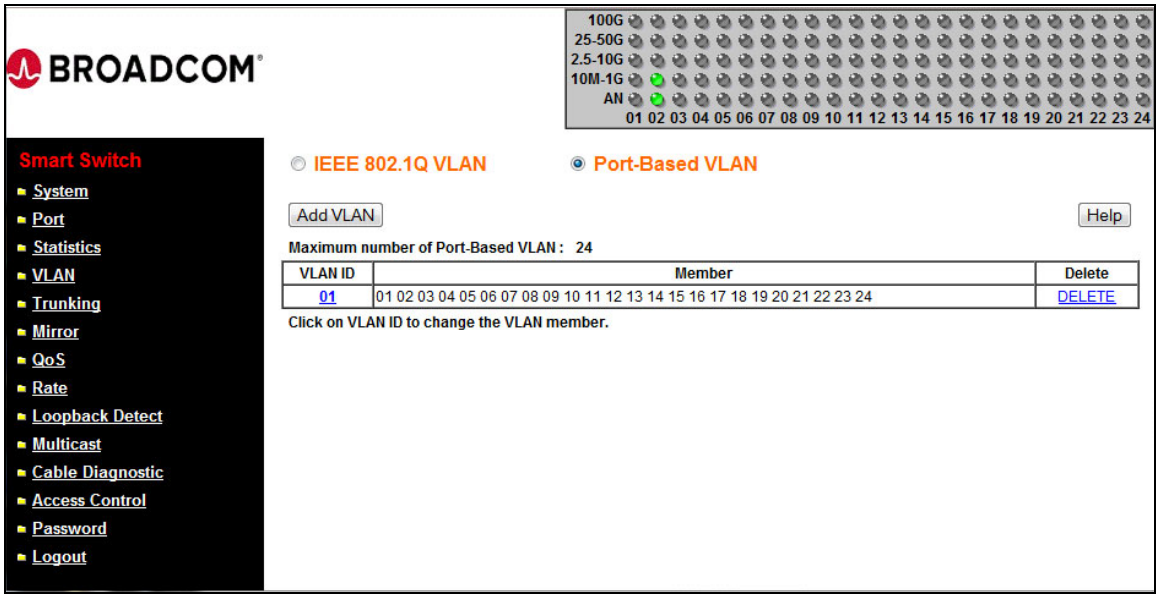
Create IEEE 802.1Q VLAN



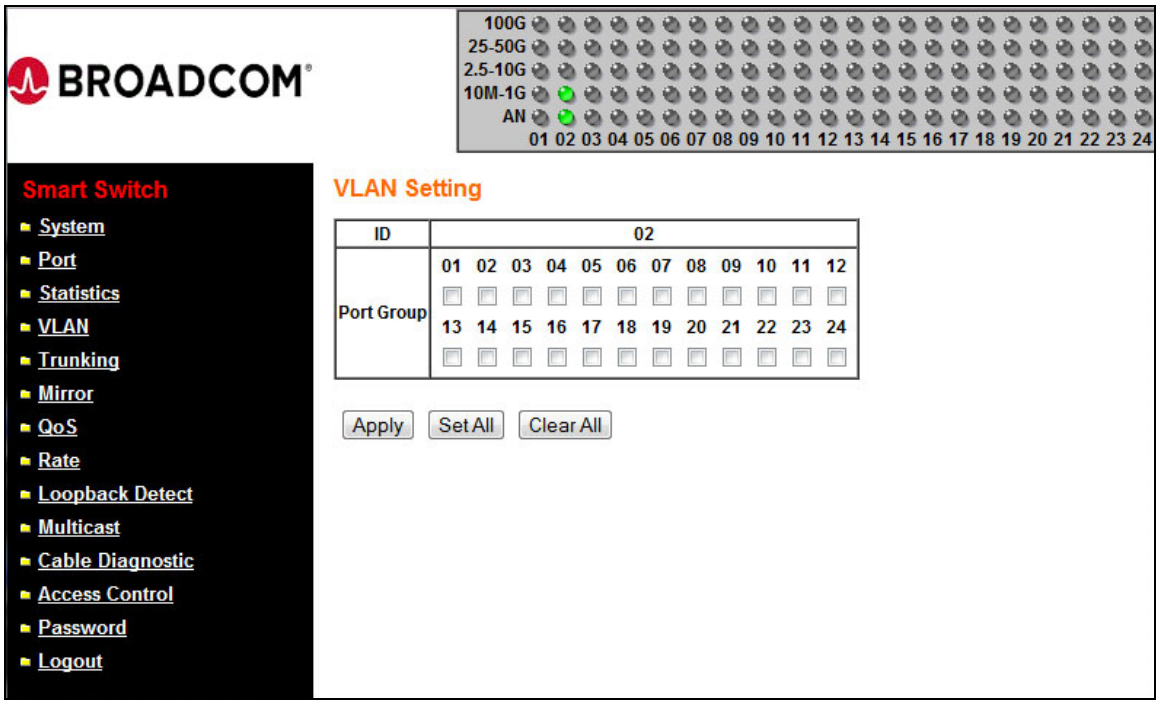
VLAN Page—Port-Based VLAN

Table 9: VLAN Page—Port-Based VLAN Parameter Description

Parameter/Button	*Description*
Add VLAN	Adds a new entry of port-based VLAN with port membership.
VLAN ID	Port-based VLAN ID.
Member	Member ports of the VLAN.
Delete	Removes the specified port-based VLAN.



Add Port-Based VLAN

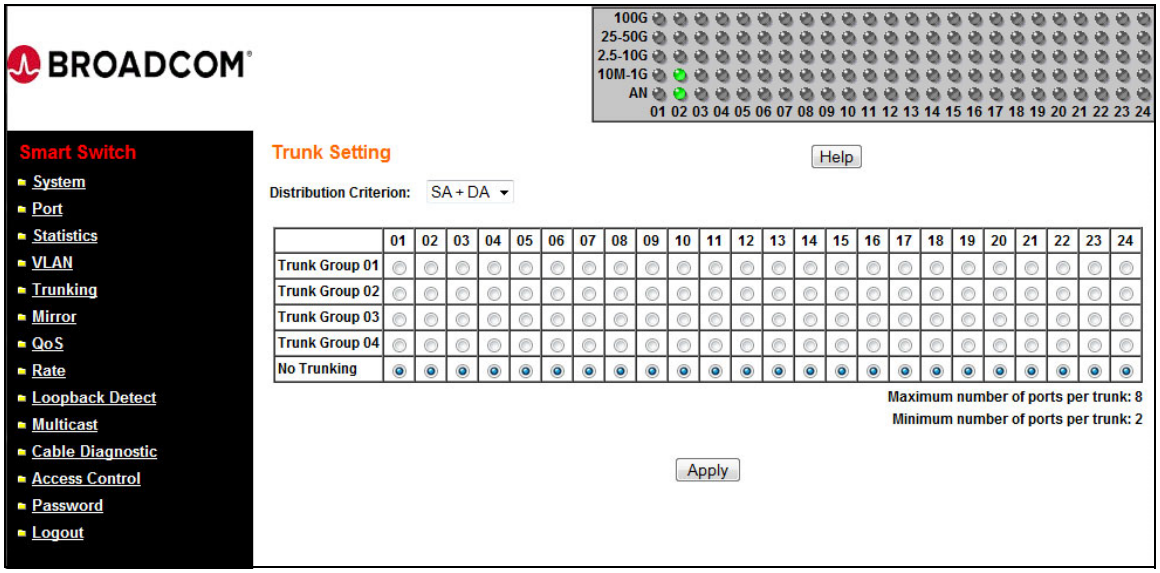


Trunk Setting Page

Trunking aggregates multiple ports into a trunk. It uses a distribution algorithm to balance traffic between trunk members.

Table 10: Trunk Setting Page Parameter Description

Parameter/Button	*Description*
Distribution Criterion	Defines the traffic-distribution algorithm between trunk member ports.
Trunk Group	Trunk Group ID supported in the device.
Member Ports	Defines member ports of the trunk.

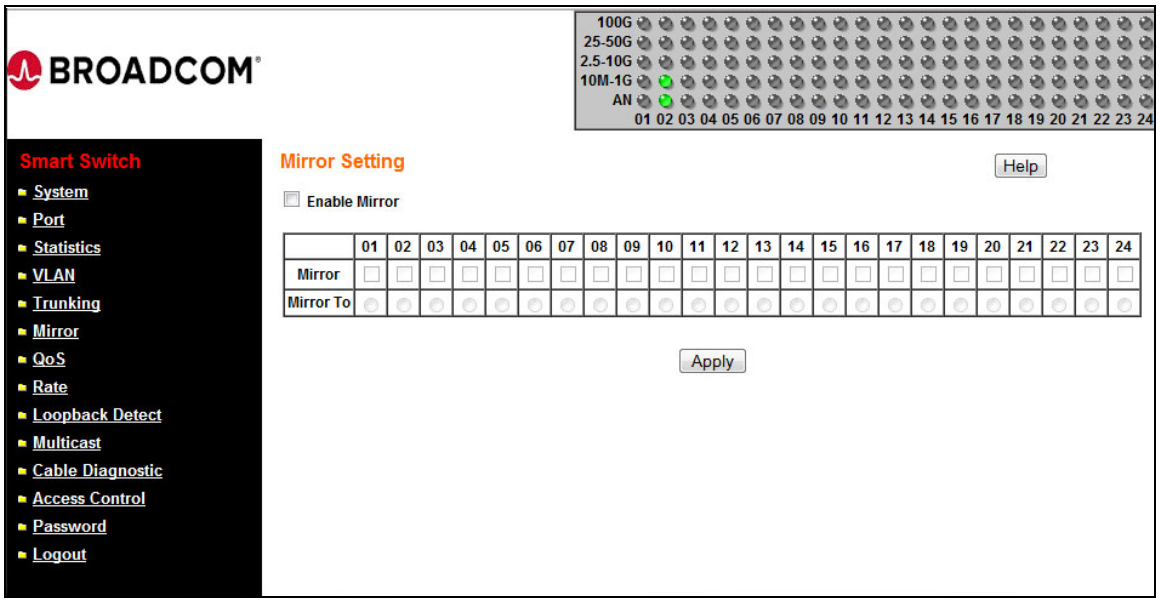


Mirror Setting Page

Mirroring monitors traffic from some given ports to a single mirror-to port. Ingress and/or egress traffic is copied from the mirroring port to the mirror-to port.

Table 11: Mirror Setting Page Parameter Description

Parameter/Button	*Description*
Enable Mirror	Enables mirroring when selected.
Mirror	Specifies an ingress/egress mirror port to which ingress/egress traffic will be mirrored.
Mirror To	Specifies the mirrored-to port.
Apply	Applies the mirror setting to the system.

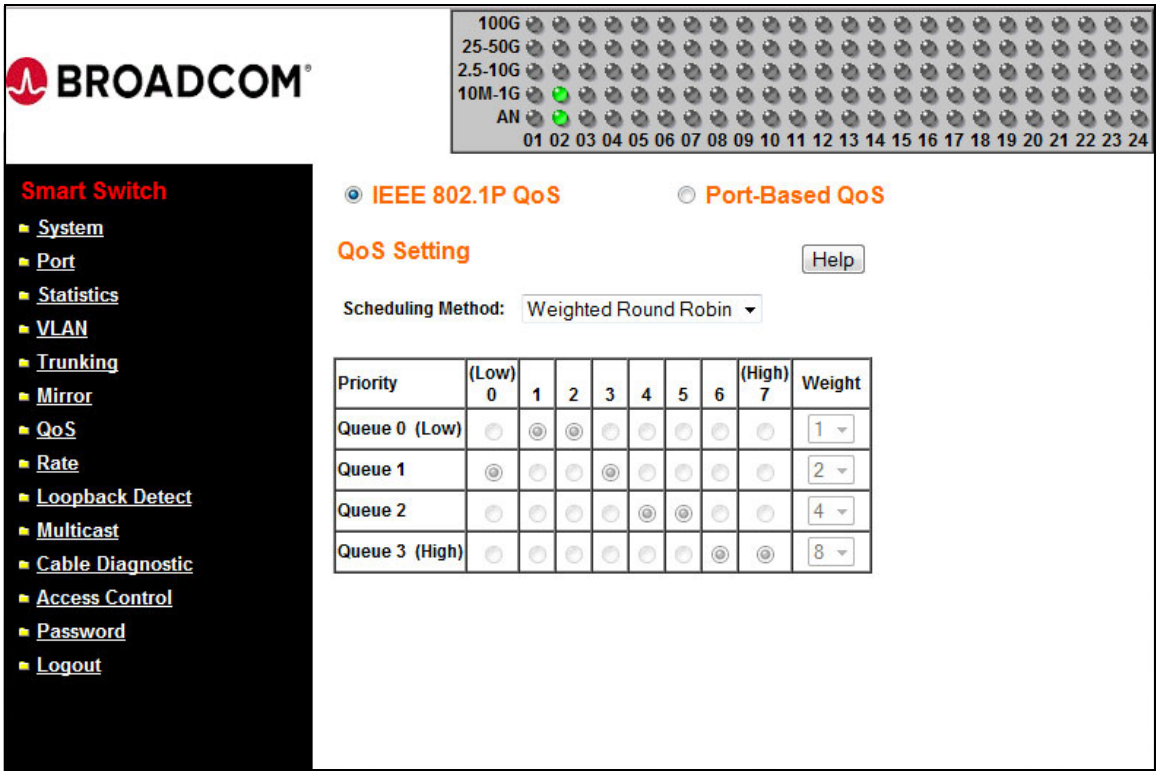


QoS Setting Page-IEEE 802.1P QoS

IEEE 802.1P QoS operation sets the queues and priority relationship. Currently, UM only supports the default IEEE 802.1P setting value.

Table 12: QoS Setting Page—IEEE 802.1P QoS Parameter Description

Parameter/Button	*Description*
Scheduling Method	Specifies one scheduling method (Weighted Round Robin) for the queues.
Queue [0:3]	Specifies four queues. Queue 0 is the lowest-priority queue and Queue 3 is the highest-priority queue. Packets in queue 3 are served more often than packets in queue 0.
Priority	Indicates packet priority. This value is retrieved from the priority tag field, with values from 0 to 7. Click on the radio button to send packets of a specific priority to a particular queue.
Weight	Indicates the weight (number of packets) to be served in the queue before moving to serve the next queue. A high-priority queue should have a higher weight than a low-priority queue.

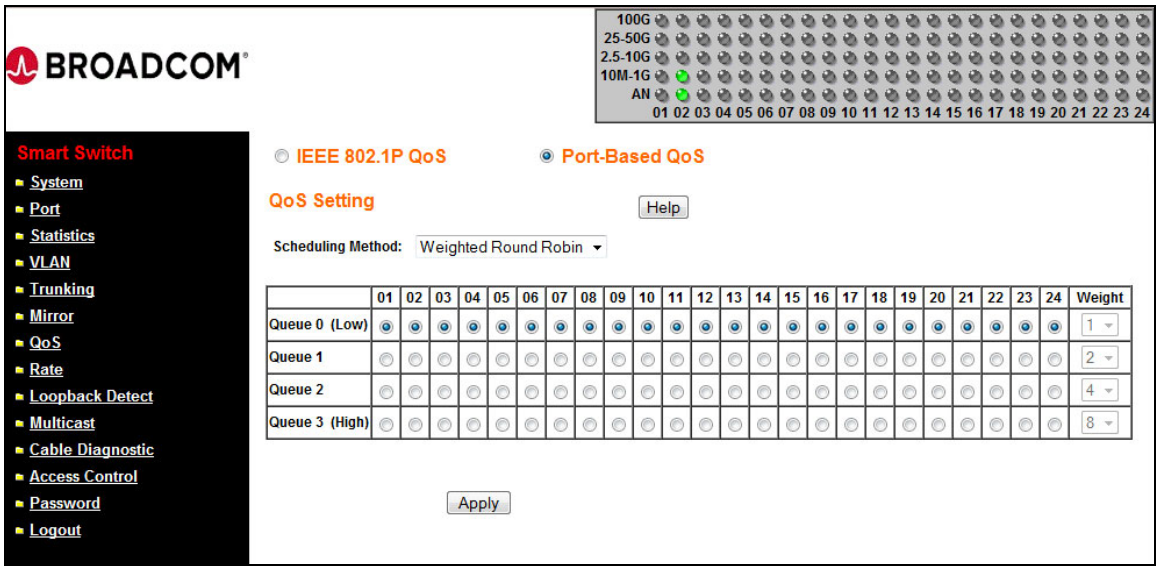


QoS Setting Page-Port-Based QoS

Port-Base QoS operation sets the port and queue relationship, selects the scheduling method for these queues and {1, 2, 4, 8} weight for mapping from low to high queue.

Table 13: QoS Setting Page—Port-Based QoS Parameter Description

Parameter/Button	*Description*
Scheduling Method	Specifies one scheduling method (Weighted Round Robin) for the queues.
Queue [0:3]	Specifies four queues. Queue 0 is the lowest-priority queue and Queue 3 is the highest-priority queue. Packets in queue 3 are served more often than packets in queue 0.
Ports	Indicates port number.
Weight	Indicates the weight (number of packets) to be served in the queue before moving to serve the next queue. A high-priority queue should have a higher weight than a low-priority queue.

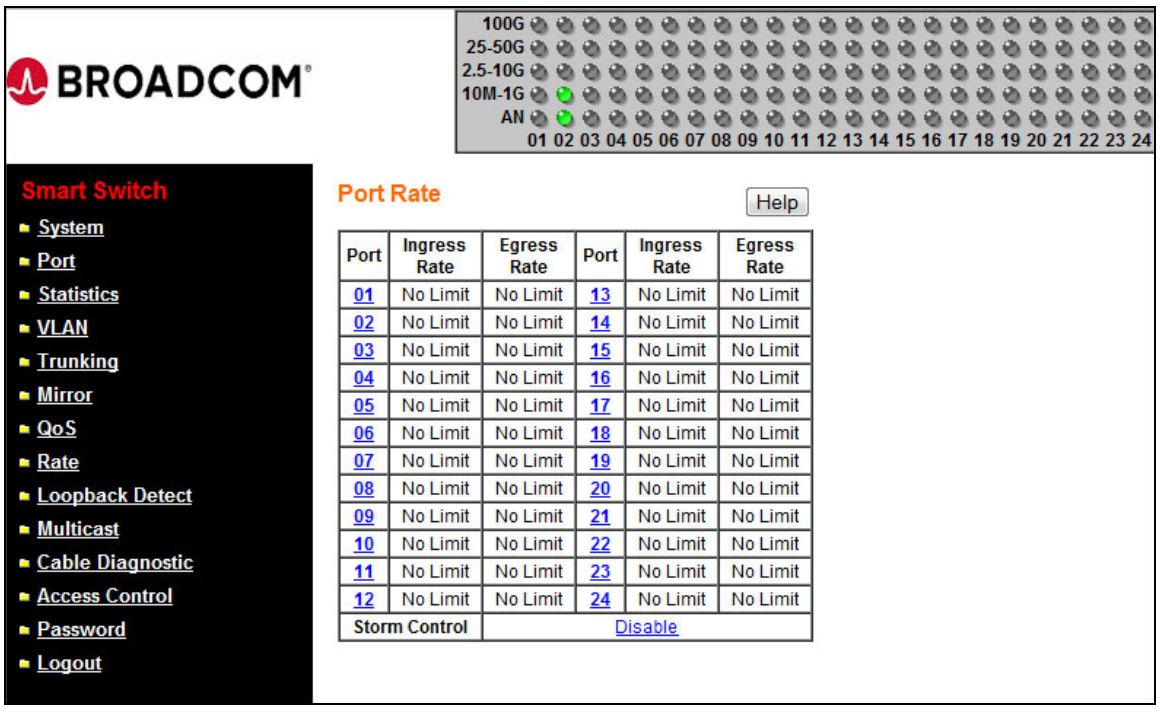


Port Rate Page

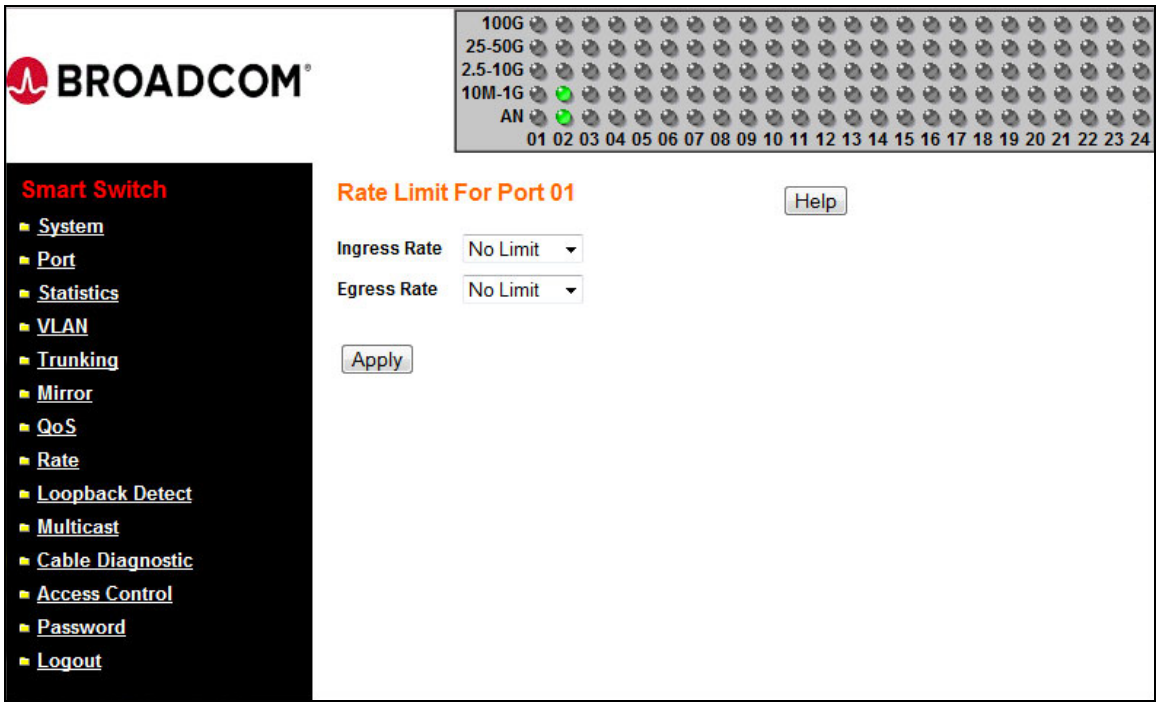
The Port Rate page is used for rate limit and storm control.

Table 14: Port Rate Page Parameter Description

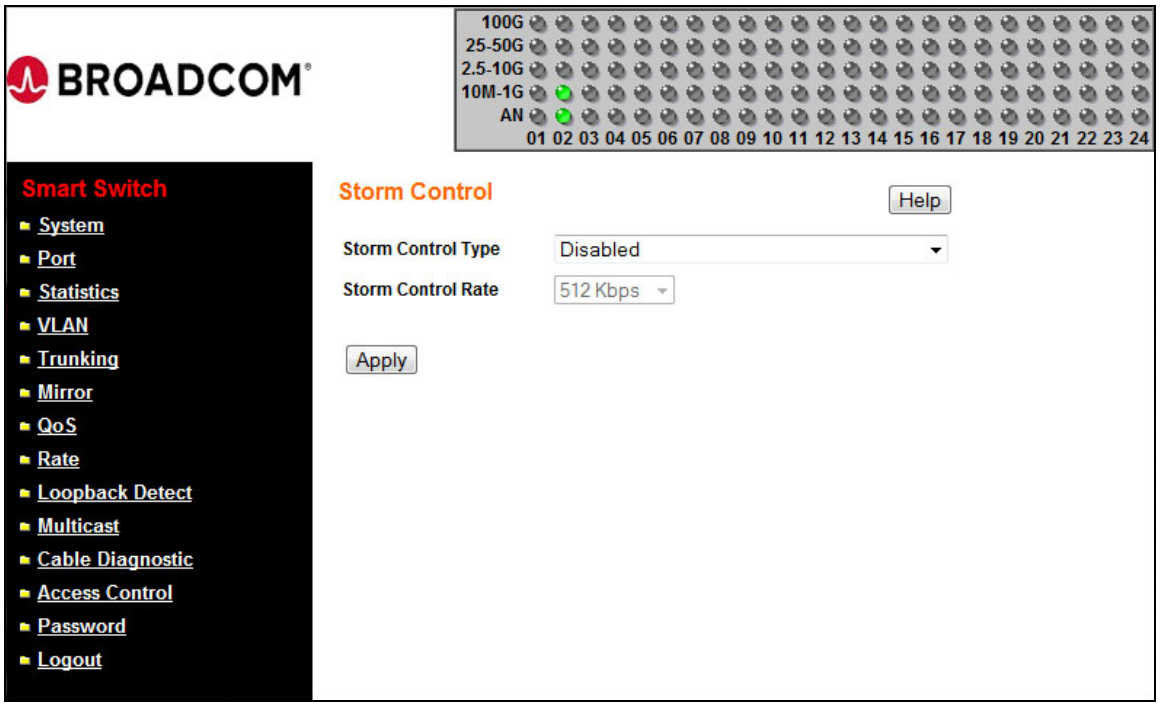
Parameter/Button	*Description*
Rate Control	Bandwidth of ingress and egress traffic for a given port.
Port	Port number. Click the port number to control the ingress and egress rates for the port. See “Per Port Rate Limit” on page 62.
Ingress Rate	Rate limitation of incoming traffic in this port. See “Per Port Rate Limit” on page 62.
Egress Rate	Rate limitation of outgoing traffic in this port. See“Per Port Rate Limit” on page 62.
Storm Control	Storm control rate. See “Per System Storm Control” on page 63.



Per Port Rate Limit

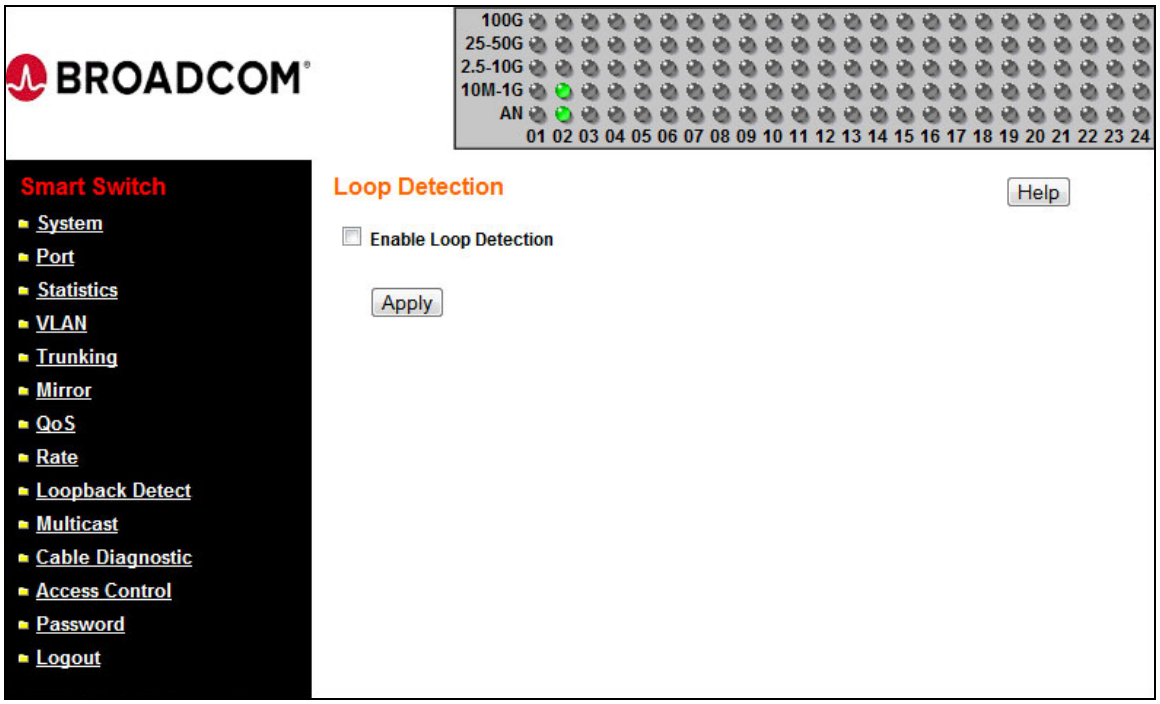


Per System Storm Control



Loop Detection Page

Select Loop Detection to enable loop traffic detection.

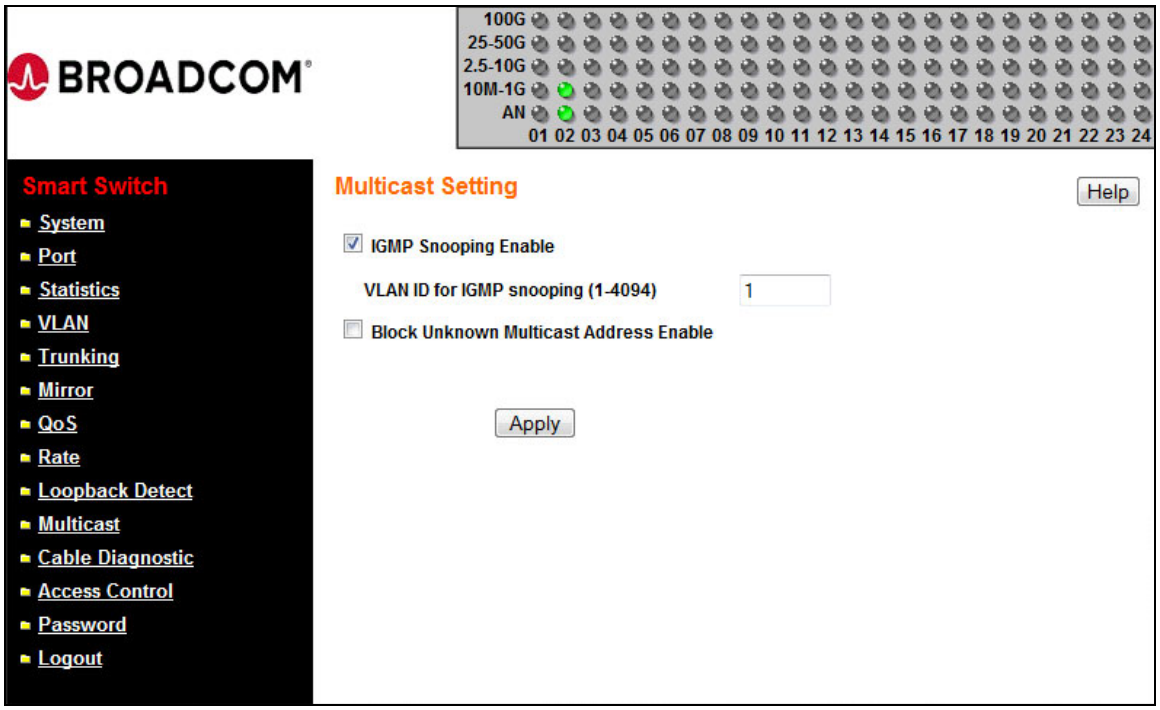


Multicast Page

IGMP is a standard defined in RFC1112 for IGMPv1, and in RFC2236 for IGMPv2. IGMP specifies how a host can register to receive specific multicast traffic from a multicast server. IGMP snooping can reduce multicast traffic at Layer 2 by configuring Layer 2 LAN ports dynamically to forward multicast traffic only to those ports that are configured to receive it.

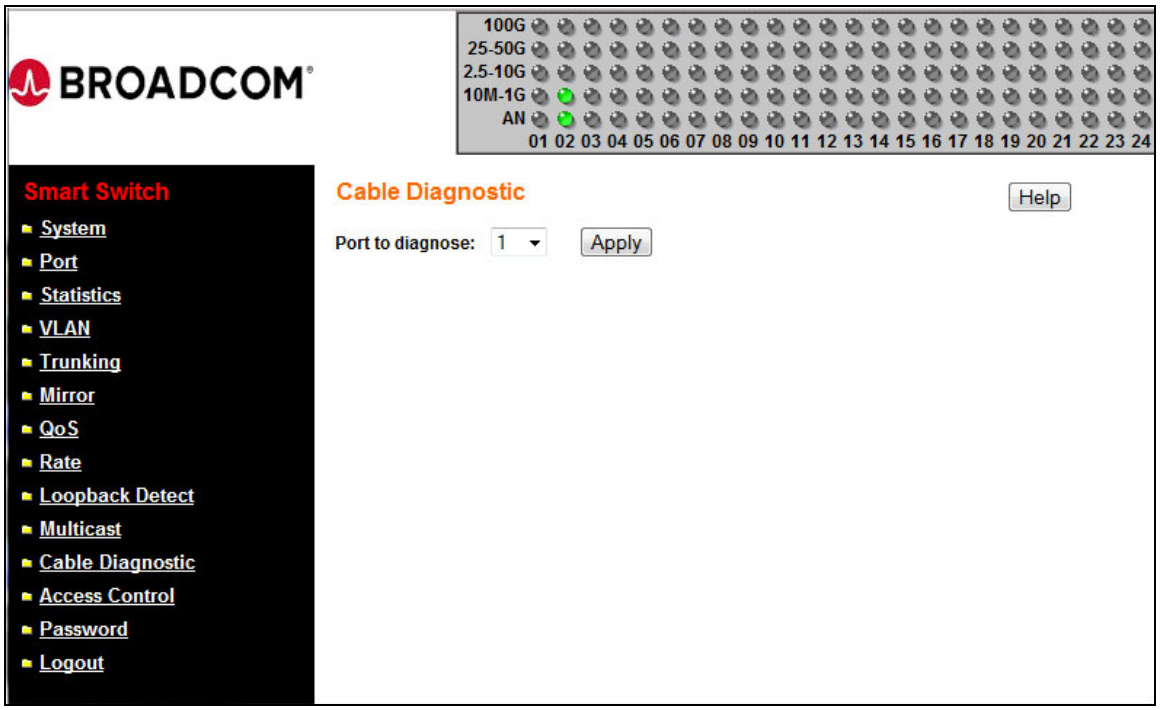
Table 15: Multicast Page Parameter Description

Parameter/Button	*Description*
IGMP Snooping Enable	Enables IGMP snooping when selected.
Block Unknown Multicast Address Enable	Enable blocking of unknown multicast address when selected.
VLAN ID for IGMP Snooping	UM only allows customers to enable IGMP snooping on one VLAN.
Apply	Applies the multicast setting to the device.



Cable Diagnostic Page

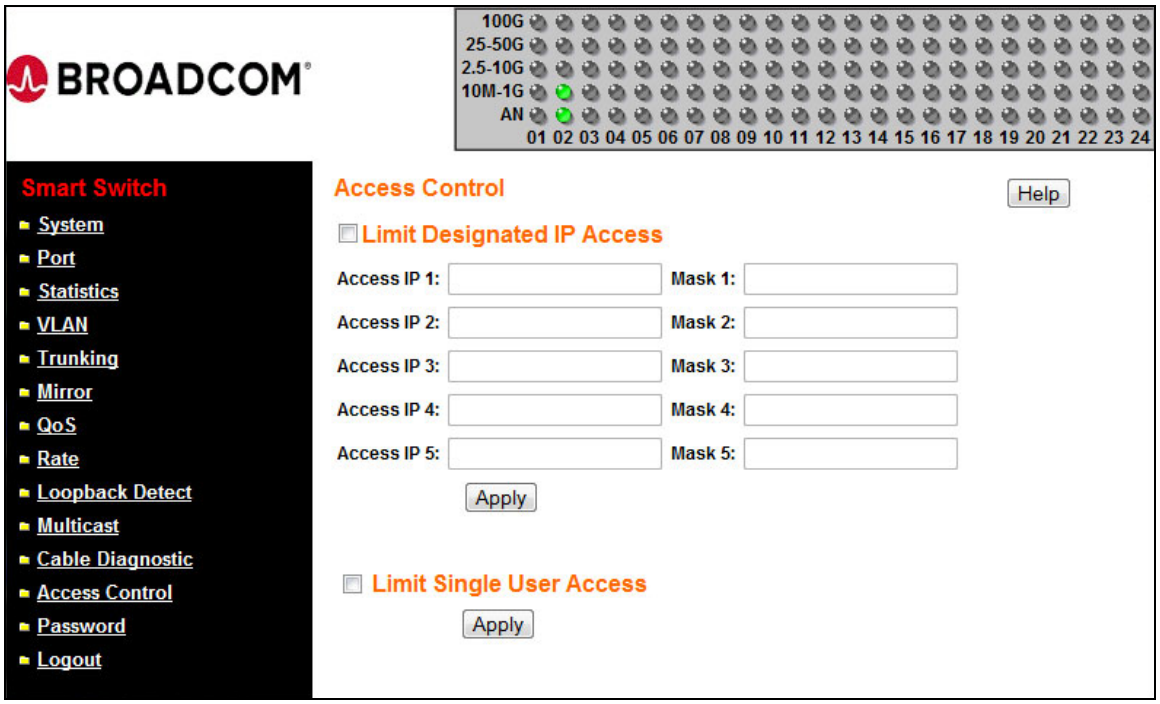
The Cable Diagnostic page provides cable status (ok/open/no cable), and distance from the failure point if it is an open cable.



Access Control Page

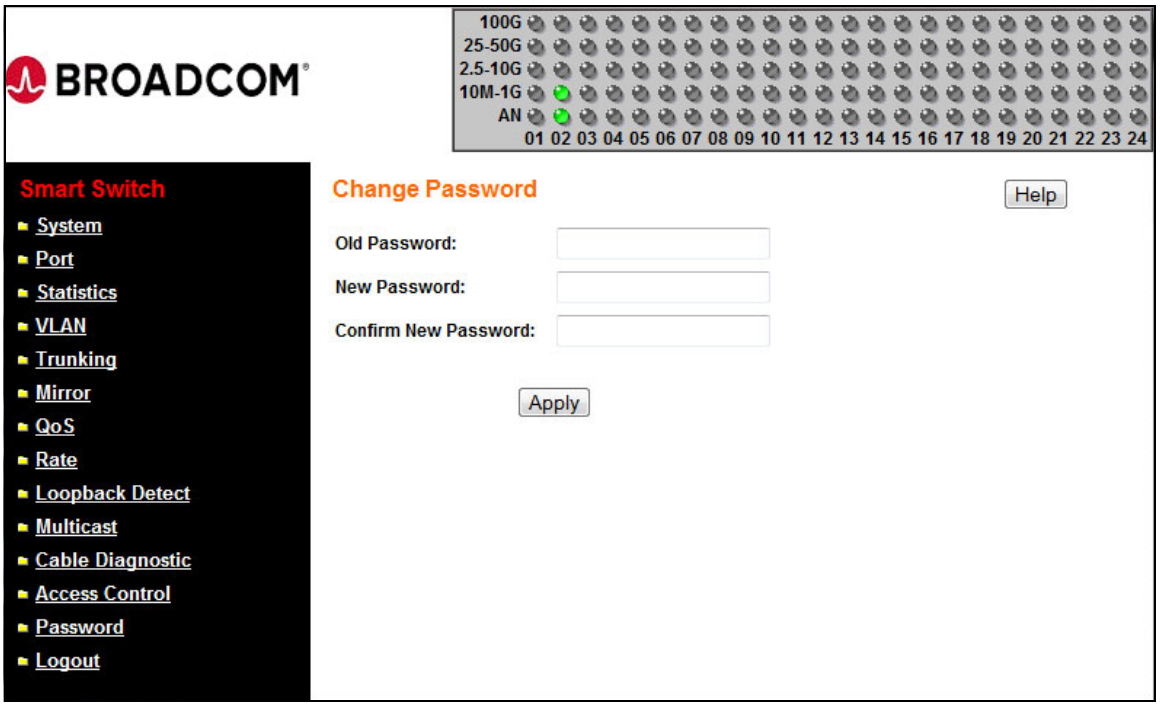
You can use this page to limit access to the web GUI to a designated source IP address and to a single user only.

For **Limit Designated IP Access**, you can assign five sets of source IP addresses/masks.



Password Page

The maximum length of a password is 20 characters. All printable characters are allowed.



Web: www.broadcom.com
Corporate Headquarters: San Jose, CA

Unmanaged-SWUM104-R May 19, 2016