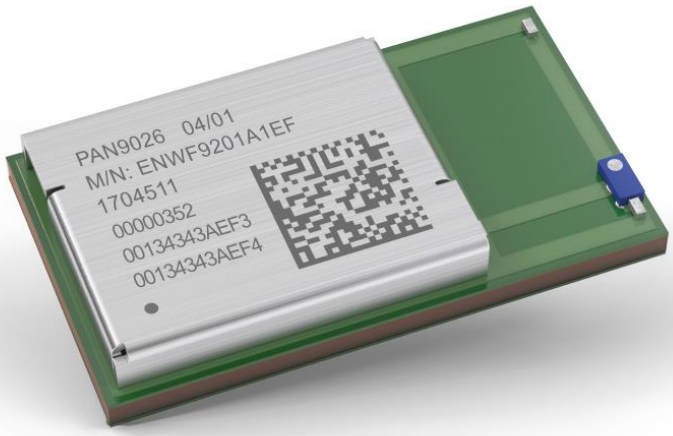
Panasonic

PAN9026

Wi-Fi Dual Band 2.4/5 GHz and Bluetooth® Module

Software Guide

Rev. 1.0



Wireless Modules

By purchase of any of the products described in this document the customer accepts the document's validity and declares their agreement and understanding of its contents and recommendations. Panasonic Industrial Devices Europe GmbH (Panasonic) reserves the right to make changes as required at any time without notification. Please consult the most recently issued User Guide before initiating or completing a design.

This User Guide does not lodge the claim to be complete and free of mistakes.

Engineering Samples (ES)

If Engineering Samples are delivered to the customer, these samples have the status “Engineering Samples”. This means that the design of this product is not yet concluded. Engineering Samples may be partially or fully functional, and they may differ from the published Product Specification.

Engineering Samples are not qualified and they are not to be used for reliability testing or series production.

Disclaimer

The customer acknowledges that samples may deviate from the User Guide and may bear defects due to their status of development and the lack of qualification mentioned above.

Panasonic rejects any liability or product warranty for Engineering Samples. In particular, Panasonic disclaims liability for damages caused by:

- The use of the Engineering Sample other than for evaluation purposes, particularly the installation or integration in another product to be sold by the customer,
- Deviation or lapse in function of the Engineering Sample,
- Improper use of the Engineering Sample.

Panasonic Industrial Devices Europe GmbH disclaims any liability for consequential and incidental damages. In case of any queries regarding the Engineering Samples, please contact your local sales partner or the related product manager.

Any outlined or referenced (source) code within this document is provided on an “as is” basis without any right to technical support or updates and without warranty of any kind on a free of charge basis according to § 516 German Civil Law (BGB) including without limitation, any warranties or conditions of title, non-infringement, merchantability, or fitness for a particular purpose. Customer acknowledges that (source) code may bear defects and errors.

The third-party tools mentioned in this document are offered by independent third-party providers who are solely responsible for these products. Panasonic has no responsibility whatsoever for the performance, product descriptions, specifications, referenced content, or any and all claims or representations of these third-party providers. Panasonic makes no warranty whatsoever, neither express nor implied, with respect to the goods, the referenced contents, or any and all claims or representations of the third-party providers.

To the maximum extent allowable by Law Panasonic assumes no liability whatsoever including without limitation, indirect, consequential, special, or incidental damages or loss, including without limitation loss of profits, loss of opportunities, business interruption, and loss of data.

Table of Contents

1 About This Document

1.1 Purpose and Audience

This Software Guide describes the basic steps required for compiling and installing the software drivers, bringing up and configuring the PAN9026 on a standard Linux® distribution. It further describes the basic usage of the Wireless Utilities and gives an introduction to the command structure. The document is intended for software engineers.

The product is referred to as “the PAN9026” or “the module” within this document.

1.2 Revision History

Revision	Date	Modifications/Remarks
1.0	2019-04-11	First version

1.3 Use of Symbols

Symbol	Description
	**Note** Indicates important information for the proper use of the product. Non-observance can lead to errors.
	**Attention** Indicates important notes that, if not observed, can put the product’s functionality at risk.
	**Tip** Indicates useful information designed to facilitate working with the PAN9026.
⇨ [chapter number] [chapter title]	**Cross reference** Indicates cross references within the document. **Example:** Description of the symbols used in this document ⇨ 1.3 Use of Symbols.
✓	**Requirement** Indicates a requirement that must be met before the corresponding tasks can be completed.
➔	**Result** Indicates the result of a task or the result of a series of tasks.
This font	**GUI text** Indicates fixed terms and text of the graphical user interface. **Example:** Click **Save**.

Symbol	Description
Menu > **Menu item**	**Path** Indicates a path, e.g. to access a dialog. **Example:** In the menu, select **File** > **Setup page**.
`This font`	**File names, messages, user input** Indicates file names or messages and information displayed on the screen or to be selected or entered by the user. **Examples:** `pan9026.c` contains the actual module initialization. The message `Failed to save your data` is displayed. Enter the value `Product 123`.

1.4 Related Documents

Please refer to the Panasonic website for related documents ⇨ 10.2 Product Information.

2 Overview

The PAN9026 is a dual band 2.4/5 GHz 802.11 a/b/g/n Wi-Fi radio module with integrated Bluetooth BR/EDR/Low Energy, specifically designed for highly integrated and cost-effective applications. The simultaneous and independent operation of the two standards enables high data rates (802.11n) and low-power operation (Bluetooth Low Energy).

A Linux-based installation has been chosen as the software platform. The Linux kernel provides an established environment for running the Wi-Fi and Bluetooth drivers and the available applications make it possible to use the PAN9026 module to its full extend.

Please refer to the Panasonic website for related documents ⇨ 10.2 Product Information.

3 Software

The PAN9026 module is based on the Marvell® chipset 88W8977 supporting 802.11a/b/g/n simultaneous station, access point, and Wi-Fi Direct operations. The module is connected through the SDIO device interface to the host processor platform running a Linux operation system. Marvell® usually provides software package releases with the driver sources and firmware binaries for a reference platform and operating system version (e.g. x86 platforms and Ubuntu® 16). The driver sources can be easily ported to comparable platforms (e.g. ARM Cortex-A) using different Linux distributions (e.g. Yocto Linux).

3.1 Linux Kernel and User Space Architecture

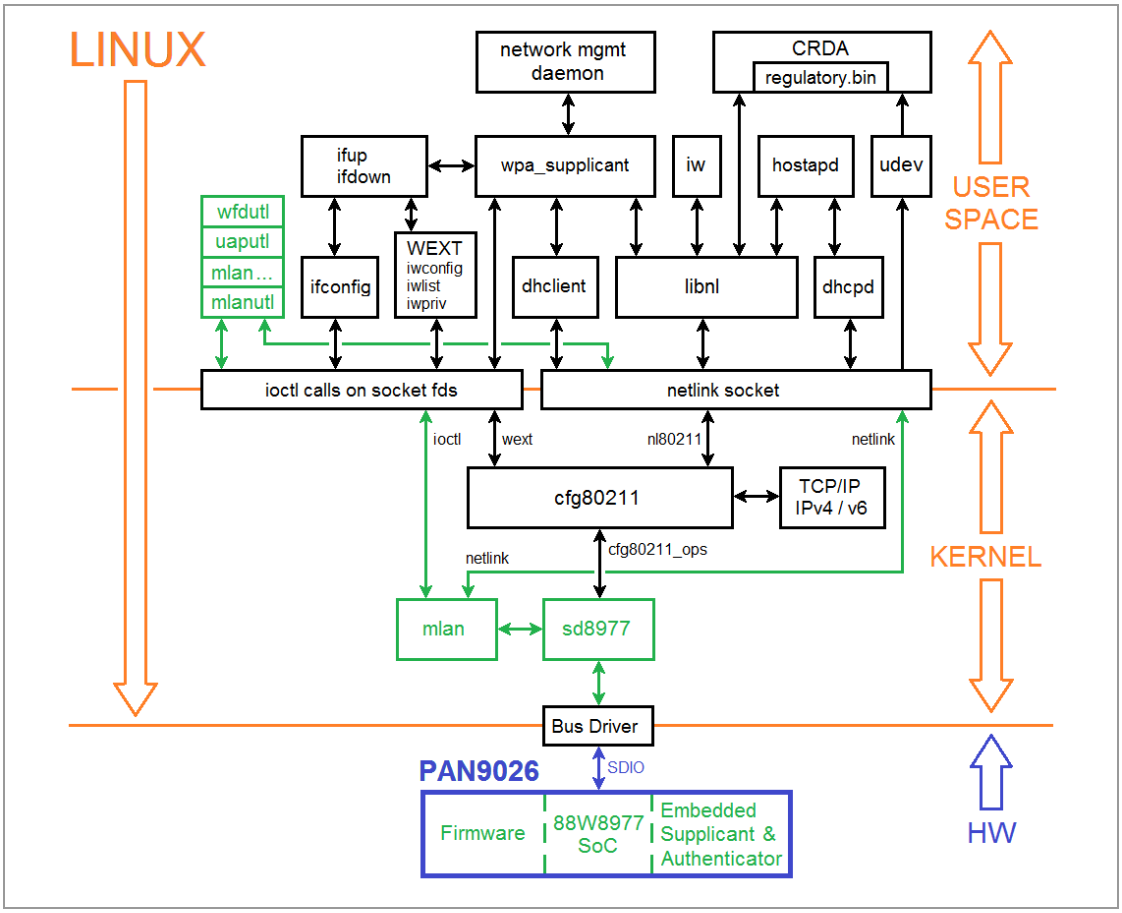
The Linux system architecture is divided into the Kernel space and user space. The Kernel runs in the dedicated part of memory. Role of the Kernel space is to manage applications and processes running in user space. It can access all the memory. If a process performs a system call, a software interrupt is sent to Kernel which then dispatches an appropriate interrupt handler. The user space is set of locations where normal user processes run. These processes cannot access Kernel space directly. Some part of Kernel space can be accessed via system calls. These system calls acts as software interrupts in Kernel space.

3.2 Linux Wireless Subsystem

The Linux Wireless (IEEE802.11) subsystem consists of the 802.11 drivers, the core mac80211 and “cfg80211” components along with the user space and in-Kernel “nl80211” configuration interface. The cfg80211 Kernel module is the Linux 802.11 configuration API. It replaces the Wireless Extensions (WExt). The “nl80211” is used to configure a “cfg80211” device and is used for Kernel to user space communication. However when implementing a “cfg80211” driver wireless extensions support can still be provided through “cfg80211” and by enabling WExt support. The “cfg80211” also provides full regulatory support; this is done through “wireless-regdb” and the usage of Central Regulatory Domain Agent (CRDA). It is a user space agent which uploads regulatory domains into the Kernel and it acts as an “udev” helper. All Linux wireless drivers should be written targeting either “cfg80211” for FullMAC devices or “mac80211” for SoftMAC devices.

3.3 Marvell Drivers and Tools in the Wireless Subsystem

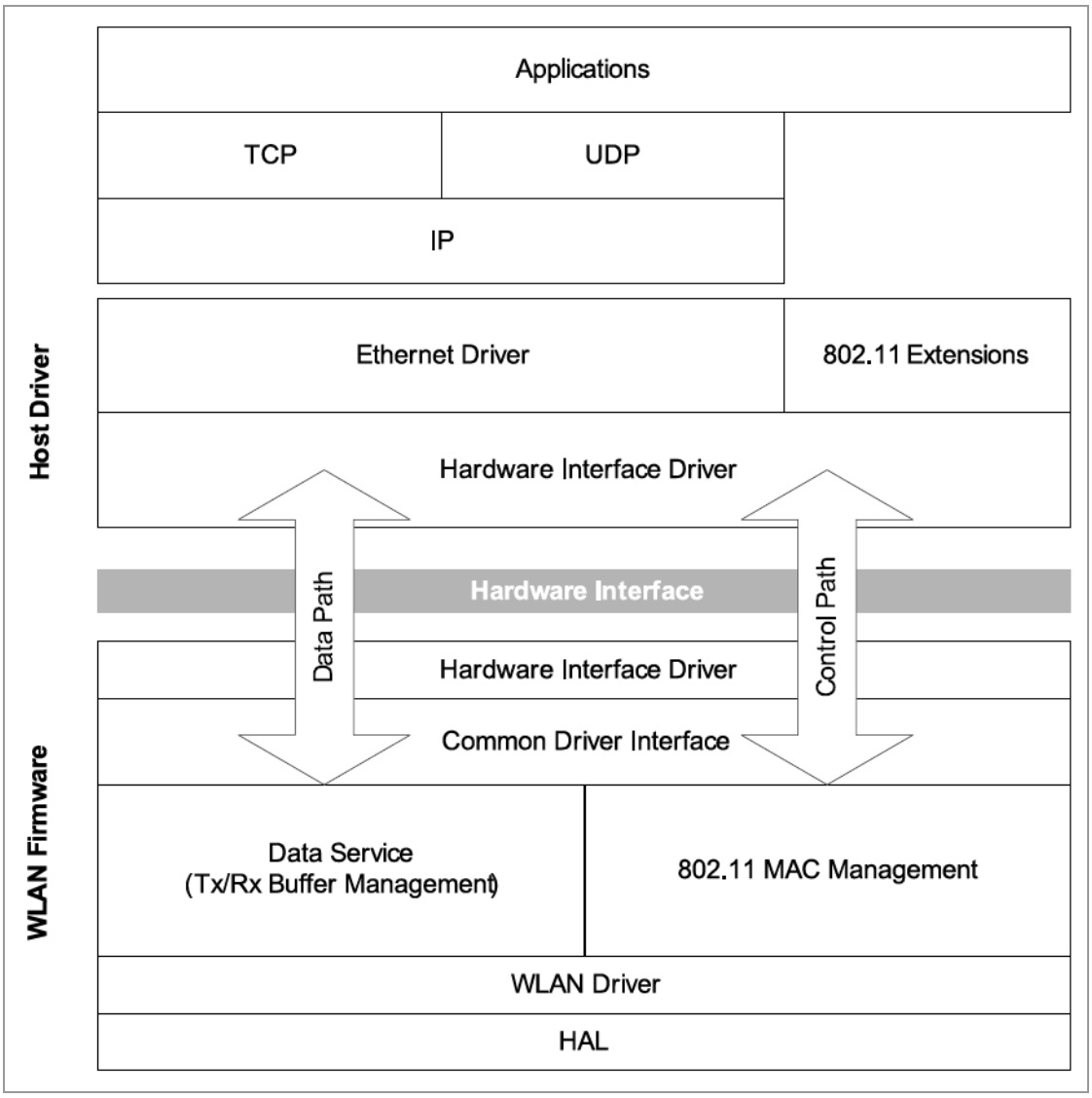
The PAN9026 module requires a Kernel driver placed on the host system and a firmware running on the module's 88W8977 System-on-Chip. The firmware image is downloaded to the module if the bus driver detects the module's SDIO interface. The Marvell® Kernel driver (sd8977) is placed between the bus driver and the attached network stack from the "cfg80211" Kernel module, which is part of the Linux Wireless (IEEE802.11) subsystem. Additionally Marvell® provides various control and configuration tools, which are located in the Linux User Space. These tools are using either ioctl (input-output control) calls or the netlink socket.



3.4 Driver and Firmware Architecture

The Marvell® drivers are installed in the Kernel space of the Linux host operating system. The WLAN device firmware image is stored on the host system as well, but it is downloaded through the bus driver interface to the module during loading of host driver. After hardware initialization the firmware is running on the module and network devices are registered on the host system.

The architecture of the host driver and WLAN device firmware is shown below. The firmware is typical of a thick firmware architecture, where it handles all 802.11 MAC management tasks. The host driver downloads standard 802.3 frames to the firmware to transmit over the wireless link as 802.11 frames. Further the WLAN firmware processes the received 802.11 frames and converts them into 802.3 frames before forwarding to the host driver.



Installation steps by host driver

1. The host driver registers the Kernel module with the SDIO Bus driver (mmc device).
2. After successful registration the bus driver calls the driver's probe handler after having detected the module hardware.
3. The probe handler allocating and initializing internal structures registers the interrupt service and starts the driver threads.
4. The firmware image is downloaded to the module.
5. After successful firmware download the module hardware is initialized.
6. Ethernet network devices (mlan0, uap0, wfd0, nan0) are created based on driver mode.

4 Preparation of Software Drivers

4.1 Download Software Package

The software package with the Marvell® driver sources and firmware binaries can be downloaded from the Marvell Extranet https://extranet.marvell.com/login/index.jsp (**My Products** > **Wireless** > **88W8977** > **Software** > **Linux** > …).

Please contact your local Panasonic Sales office for creating a customer account to get access to the Marvell Extranet.

Linux Mainline Kernel Support

Linux mainline Kernels contain the IEEE 802.11n SDIO/PCI-E/USB FullMAC driver “mwifiex”.

The supported Marvell chipsets can be found at the Linux Wireless Wiki page https://wireless.wiki.kernel.org/.

The Marvell chipset “88W8977” used in PAN9026 is not supported in mainline. It is necessary to download the software package from the Marvell Extranet.

4.2 Software Package Versions

For chipset “88W8977” there are following software package versions available at the Marvell Extranet, which are belonging to the host communication interface and software license type:

Software Package Version	WLAN Interface	Bluetooth Interface	Software License
SD-WLAN-SD-BT-8977-…MGPL.zip	SDIO	SDIO	MGPL (full)
SD-WLAN-UART-BT-8977-…-MGPL.zip	SDIO	UART	MGPL (full)
SD-WLAN-SD-BT-8977-…-GPL.zip	SDIO	SDIO	GPL
SD-WLAN-UART-BT-8977-…-GPL.zip	SDIO	UART	GPL

Software License Type

GPL Release

The module “moal” is licensed under the GPL, the module “mlan” and “mapps” sample applications are Marvell Proprietary Code.

MGPL Release

The modules “moal” and “mlan”, and “mapps” sample applications are all licensed under the GPL.

Supported Linux Kernel Version

Linux Kernel Version	From	To	FW Version
Software Package Release	2.6.34	4.19	W16.68.1.p195-16.26.1.p195

Host Communication Interface

The PAN9026 has two Host Communication Interfaces. First an SDIO device interface that conforms to the industry standard SDIO full-speed card specification and allows a host controller using the SDIO bus protocol to access the device. The device supports SDIO 3.0 Standard with 1-bit SDIO or 4-bit SDIO transfer modes with full clock range up to 100 MHz. Second a high-speed Universal Asynchronous Receiver/Transmitter (UART) interface, compliant with the “industry standard 16550 specification”.

The SDIO interface is used for the communication between WLAN Kernel driver (“sd8977”) and firmware. It is also used for uploading the firmware during the WLAN Kernel driver loading. This interface can also be used for the Bluetooth HCI communication between Bluetooth Kernel driver (“bt8977”, “mbt8977”) and firmware. In this case it is a time-shared interface for WLAN and Bluetooth communication. The UART interface can be separately used for the Bluetooth HCI communication between Bluetooth Kernel driver (“bt8977”, “mbt8977”) and firmware. It is offloading the SDIO and internal bus traffic for WLAN processing.

Software Package Release Version Information Example

Example:

`SD-WLAN-SD-BT-8977-U16-MMC-W16.68.1.p195-16.26.1.p195-C4X16625_V2-MGPL`

Type	Convention	Description
WLAN Interface Prefix	SD-WLAN	SDIO interface for WLAN device communication
Bluetooth Interface Prefix	SD-BT (UART-BT)	SDIO or UART interface for Bluetooth HCI communication
Marvell SoC Version	8977	Chipset 88W8977 (PAN9026)
Linux Distribution	U16	Ubuntu 16 Test Environment
Interface Type	MMC	Multimedia Card

Type	Convention	Description
WLAN Firmware Version	W16.68.1.p195	WLAN firmware: `sd8977_wlan_v2.bin`
	W	"W" indicates the release includes the WPA/WPA2 VU, WPA Replay Protection Fixes required for WFA Security Detection/Certification testing purpose and Memory copy vulnerability fix.
	16	Major Revision (first number from the left): Tracks the main firmware version
	68	Minor Revision (second number from the left): Tracks the chip family, firmware branch, custom projects, etc.
	1	Release Number (third number from the left): Tracks the incremental changes in the consequent firmware releases given to Quality Assurance (QA) or customers.
	p195	Patch Number (fourth number from the left): The patch number starts at zero (no patch) and increments with release of subsequent builds with bug fixes.
Bluetooth Firmware Version	16.26.1.p195	Bluetooth firmware: `sd8977_bt_v2.bin`, `uart8977_bt_v2.bin`
	16	Major Revision (first number from the left)
	26	Minor Revision (second number from the left)
	1	Release Number (third number from the left)
	p195	Patch Number (fourth number from the left)
Driver Package Version	C4X16625	WLAN Driver: `mlan.ko`, `sd8977.ko` SD-BT Driver: `bt8977.ko` (BlueZ), `mbt8977.ko` (HCI) UART-BT Driver: `hci_uart.ko`
	C	Indicates Marvell WLAN/Bluetooth combo driver: `sdsd8977_combo_v2.bin`, `sduart8977_combo_v2.bin`
	4X	Indicates the version of Linux kernel used to release the Marvell driver.
	16	Release Number: Tracks the incremental changes in the consequent driver releases given to QA.
	625	Patch Number: The patch number starts at zero (no patch) and increments with release of subsequent builds with more bug fixes.
Firmware Version Type	V2	V2: 88W8977 SoC – A1 Revision
Software License Type	MGPL	Marvell full GPL Code ("moal"," mlan", and "mapps")

4.3 Unpack Software Package

1. Create a working directory for the software package ZIP archive as following:

```
$> mkdir -p /tmp/pan9026
$> cd /tmp/pan9026
$> TOP=$(pwd)
$> cd ${TOP}
```

➔ In the directory `tmp` a working directory `pan9026` is created.

➔ After having changed to the subdirectory `pan9026` a variable `TOP` is generated as actual working directory.

2. Copy the software package ZIP archive to working directory as following:

```
$> cp [path-to-ZIP]/SD-WLAN-SD-BT-8977-U16-MMC-W16.68.1.p195-
16.26.1.p195-C4X16625_V2-MGPL.zip ${TOP}
```

3. Unzip the software package ZIP archive inside working directory as following:

```
$> cd ${TOP}
$> unzip SD-WLAN-SD-BT-8977-U16-MMC-W16.68.1.p195-16.26.1.p195-
C4X16625_V2-MGPL.zip -d p195
Archive:  SD-WLAN-SD-BT-8977-U16-MMC-W16.68.1.p195-16.26.1.p195-
C4X16625_V2-MGPL.zip
  inflating: SD-WLAN-SD-BT-8977-U16-MMC-W16.68.1.p195-16.26.1.p195-
C4X16625_V2-MGPL-ReleaseNotes.pdf
  inflating: SD-WLAN-SD-BT-8977-U16-MMC-W16.68.1.p195-16.26.1.p195-
C4X16625_V2-MGPL.tar
```

➔ A subdirectory with the name of the software patch has been created. The release notes and a TAR file have been inflated inside the subdirectory.

4. Change directory to the SW patch subdirectory and uncompressing the TAR files:

```
$> cd ${TOP}/p195
$> ls
SD-WLAN-SD-BT-8977-U16-MMC-W16.68.1.p195-16.26.1.p195-C4X16625_V2-
MGPL-ReleaseNotes.pdf
SD-WLAN-SD-BT-8977-U16-MMC-W16.68.1.p195-16.26.1.p195-C4X16625_V2-
MGPL.tar
$> tar xvf SD-WLAN-SD-BT-8977-U16-MMC-W16.68.1.p195-16.26.1.p195-
C4X16625_V2-MGPL.tar
SD-BT-8977-U16-MMC-16.26.1.p195-C4X14113_V2-GPL-src.tgz
SD-BT-CHAR-8977-U16-MMC-16.26.1.p195-C4X14113_V2-GPL-src.tgz
SD-UAPSTA-8977-U16-MMC-16.68.1.p195-C4X16625_V2-MGPL-src.tgz
SD-UAPSTA-8977-U16-MMC-16.68.1.p195-C4X16625_V2-app-src.tgz
SD-UAPSTA-8977-U16-MMC-16.68.1.p195-C4X16625_V2-mlan-src.tgz
```

```
FwImage/sd8977_bt_v2.bin
FwImage/sd8977_wlan_v2.bin
FwImage/sd8977_combo_v2.bin
```

➔ Five TGZ files have been inflated in the directory.

➔ An additional subdirectory with the name `FwImage` has been created.

5. Change directory to the SW patch subdirectory and uncompressing the TGZ files:

```
$> cd ${TOP}/p195
$> tar xvfz SD-BT-8977-U16-MMC-16.26.1.p195-C4X14113_V2-GPL-src.tgz
SD-UAPSTA-BT-8977-U16-MMC-W16.68.1.p195-16.26.1.p195-C4X16625_V2-MGPL/mbt_src/…
$> tar xvfz SD-BT-CHAR-8977-U16-MMC-16.26.1.p195-C4X14113_V2-GPL-src.tgz
SD-UAPSTA-BT-8977-U16-MMC-W16.68.1.p195-16.26.1.p195-C4X16625_V2-MGPL/mbtc_src/…
$> tar xvfz SD-UAPSTA-8977-U16-MMC-16.68.1.p195-C4X16625_V2-app-src.tgz
SD-UAPSTA-BT-8977-U16-MMC-W16.68.1.p195-16.26.1.p195-C4X16625_V2-MGPL/wlan_src/mapp/…
$> tar xvfz SD-UAPSTA-8977-U16-MMC-16.68.1.p195-C4X16625_V2-MGPL-src.tgz
SD-UAPSTA-BT-8977-U16-MMC-W16.68.1.p195-16.26.1.p195-C4X16625_V2-MGPL/wlan_src/mlinux/…
SD-UAPSTA-BT-8977-U16-MMC-W16.68.1.p195-16.26.1.p195-C4X16625_V2-MGPL/wlan_src/README…
SD-UAPSTA-BT-8977-U16-MMC-W16.68.1.p195-16.26.1.p195-C4X16625_V2-MGPL/wlan_src/Makefile
SD-UAPSTA-BT-8977-U16-MMC-W16.68.1.p195-16.26.1.p195-C4X16625_V2-MGPL/wlan_src/script/…
$> tar xvfz SD-UAPSTA-8977-U16-MMC-16.68.1.p195-C4X16625_V2-mlan-src.tgz
SD-UAPSTA-BT-8977-U16-MMC-W16.68.1.p195-16.26.1.p195-C4X16625_V2-MGPL/wlan_src/mlan/…
$> ls
FwImage
SD-UAPSTA-8977-U16-MMC-16.68.1.p195-C4X16625_V2-MGPL-src.tgz
SD-BT-8977-U16-MMC-16.26.1.p195-C4X14113_V2-GPL-src.tgz
SD-UAPSTA-8977-U16-MMC-16.68.1.p195-C4X16625_V2-mlan-src.tgz
SD-BT-CHAR-8977-U16-MMC-16.26.1.p195-C4X14113_V2-GPL-src.tgz
SD-UAPSTA-BT-8977-U16-MMC-W16.68.1.p195-16.26.1.p195-C4X16625_V2-MGPL
SD-WLAN-SD-BT-8977-U16-MMC-W16.68.1.p195-16.26.1.p195-C4X16625_V2-MGPL-ReleaseNotes.pdf
```

```
SD-UAPSTA-8977-U16-MMC-16.68.1.p195-C4X16625_V2-app-src.tgz
SD-WLAN-SD-BT-8977-U16-MMC-W16.68.1.p195-16.26.1.p195-C4X16625_V2-MGPL.tar
```

➔ An additional subdirectory with the name of the software driver package has been created.

4.4 Copy Firmware

Linux Firmware Library

The PAN9026 requires a System-on-Chip firmware to be uploaded to the Marvell chipset “88W8977”. By default the Linux Kernel loads Marvell® firmware files from the firmware path `/lib/firmware/mrvl`.

4.4.1 Copy Firmware (SD-WLAN-SD-BT)

1. Change directory to the subdirectory `FwImage` and copy firmware binaries to firmware library:

```
$> cd ${TOP}/p195/FwImage
$> ls
sd8977_bt_v2.bin
sd8977_wlan_v2.bin
sd8977_combo_v2.bin
$> sudo cp sd8977_bt_v2.bin /lib/firmware/mrvl
$> sudo cp sd8977_wlan_v2.bin /lib/firmware/mrvl
$> sudo cp sd8977_combo_v2.bin /lib/firmware/mrvl
```

➔ The System-on-Chip firmware has been copied to the Marvell® vendor specific firmware library.

4.4.2 Copy Firmware (SD-WLAN-UART-BT)

1. Change directory to the subdirectory `FwImage` and copy firmware binaries to firmware library:

```
$> cd ${TOP}/p195/FwImage
$> ls
sd8977_wlan_v2.bin
sduart8977_combo_v2.bin
uart8977_bt_v2.bin
$> sudo cp sd8977_wlan_v2.bin /lib/firmware/mrvl
$> sudo cp sduart8977_combo_v2.bin /lib/firmware/mrvl
$> sudo cp uart8977_bt_v2.bin /lib/firmware/mrvl
```

➔ The System-on-Chip firmware has been copied to the Marvell® vendor specific firmware library.

5 Compiling Drivers

5.1 Preparation of Kernel Sources

Before compiling the Kernel drivers please ensure that the Kernel is able to compile external Kernel modules.

Please note that the Marvell® drivers cannot be compiled statically into the Kernel.

5.2 Cross-Compilation

For cross-compiling the driver sources it is necessary to specify the target architecture, the cross-toolchain, and the directory with the Kernel sources to be used to build the Kernel on the target system:

```
$> make CROSS_COMPILE=${CROSS_COMPILE} KERNELDIR=${KERNEL_DIR}
ARCH=${ARCH} build
```

Definition of Variables

`ARCH` = arm
`CROSS_COMPILE` = <path to cross-compiler>

Example: `/home/xubuntu/development/pan9026/gcc-linaro-6.4.1-2017.11-x86_64_arm-linux-gnueabihf/bin/arm-linux-gnueabihf-`

`KERNEL_DIR` = <path to Kernel sources>

Example: `/home/xubuntu/development/pan9026/linux-stable/`

5.3 Compiling Bluetooth Driver Sources

5.3.1 Building Result of Driver Sources

Following Kernel modules and utilities (configuration tools) will be generated or copied by compiling the driver sources:

File Name	Description
bt8977.ko	Bluetooth HCI Kernel driver (hci0 device)
mbt8977.ko	Bluetooth character Kernel driver (mbtchar0 device)
fmapp	Utility to configure Bluetooth connections, send HCI or vendor specific commands
load	Bluetooth driver load shell script
unload	Bluetooth driver unload shell script

File Name	Description
README	Readme files as user manual for the utilities with instructions and examples
config/*	Subdirectory for various sample configuration files used by utilities

5.3.2 Compiling Bluetooth Character Driver (SD-BT)

1. Change directory to the Bluetooth character driver source subdirectory:

```
$> cd ${TOP}/p195/SD-UAPSTA-BT-8977-U16-MMC-W16.68.1.p195-
16.26.1.p195-C4X16625_V2-MGPL
$> ls
mbtc_src  mbt_src  wlan_src
$> cd mbtc_src
$> ls
app  config  README  bt  Makefile  script
$> make clean && make build
make – find . -name "*.o" -exec rm {} \;
find . -name "*.ko" -exec rm {} \;
find . -name ".*.cmd" -exec rm {} \;
find . -name "*.mod.c" -exec rm {} \;
find . -name "*.symvers" -exec rm {} \;
find . -name "modules.order" -exec rm {} \;
find . -name ".*.dwo" -exec rm {} \;
find . -name "*dwo" -exec rm {} \;
rm -rf .tmp_versions
make -C app/fm_app clean
…
cp -f mbt8xxx.ko ../bin_sd8977_btchar/mbt8977.ko
cp -r config ../bin_sd8977_btchar
cp -f script/sdio_mmc/* ../bin_sd8977_btchar/
cp -f README ../bin_sd8977_btchar
make -C app/fm_app build INSTALLDIR=../bin_sd8977_btchar;
…
cp -f app/fm_app/fmapp ../bin_sd8977_btchar;C /lib/modules/…
$> ls
app  config    mbt8xxx.ko     mbt8xxx.mod.o  modules.order   README
bt   Makefile  mbt8xxx.mod.c  mbt8xxx.o      Module.symvers  script
$> cd ${TOP}/p195/SD-UAPSTA-BT-8977-U16-MMC-W16.68.1.p195-
16.26.1.p195-C4X16625_V2-MGPL
$> ls
bin_sd8977_btchar  mbtc_src  mbt_src  wlan_src
```

➔ An additional subdirectory has been created. The driver and utility binaries can be found in the directory `.../bin_sd8xxx_btchar`.

5.3.3 Compiling Bluetooth HCI Driver (SD-BT)

1. Change directory to the Bluetooth driver source subdirectory:

```
$> cd ${TOP}/p195/SD-UAPSTA-BT-8977-U16-MMC-W16.68.1.p195-
16.26.1.p195-C4X16625_V2-MGPL
$> ls
bin_sd8977_btchar  mbtc_src  mbt_src  wlan_src
$> cd mbt_src
$> ls
app  bt  config  Makefile  README  script
$> make clean && make build
find . -name "*.o" -exec rm {} \;
find . -name "*.ko" -exec rm {} \;
find . -name ".*.cmd" -exec rm {} \;
find . -name "*.mod.c" -exec rm {} \;
find . -name "*.symvers" -exec rm {} \;
find . -name "modules.order" -exec rm {} \;
find . -name ".*.dwo" -exec rm {} \;
find . -name "*dwo" -exec rm {} \;
rm -rf .tmp_versions
make -C app/fm_app clean
make[1]: Entering directory …
…
cp -f bt8xxx.ko ../bin_sd8977_bt/bt8977.ko
cp -r config ../bin_sd8977_bt
cp -f script/sdio_mmc/* ../bin_sd8977_bt/
cp -f README ../bin_sd8977_bt
make -C app/fm_app build INSTALLDIR=../bin_sd8977_bt;
…
cp -f app/fm_app/fmapp ../bin_sd8977_bt;make – find . -name "*.o" -exec
rm {} \;
$> ls
app  bt8xxx.ko     bt8xxx.mod.o   config    modules.order   README
bt   bt8xxx.mod.c  mbt8xxx.o      Makefile  Module.symvers  script
$> cd ${TOP}/p195/SD-UAPSTA-BT-8977-U16-MMC-W16.68.1.p195-
16.26.1.p195-C4X16625_V2-MGPL
$> ls
```

```
bin_sd8977_bt  bin_sd8977_btchar  mbtc_src  mbt_src  wlan_src
```

➔ An additional subdirectory has been created. The driver and utility binaries can be found in the directory `.../bin_sd8xxx_bt`.

5.3.4 Compiling Bluetooth HCI Driver (UART-BT)

The software package Version "SD-WLAN-UART-BT-8977-…" is released separately. The source code contents and firmware binaries are different ⇨ 4.2 Software Package Versions.

1. Change directory to the Bluetooth driver source subdirectory:

```
$> cd ${TOP}/p195/SD-UAPSTA-BT-UART-8977-U16-MMC-W16.68.1.p195-
16.26.1.p195-C4X16625_V2-MGPL
$> ls
muart_src  wlan_src
$> cd muart_src
$> ls
bt_drv.h    hci_ldisc.c  hci_uart.h     include     mbt_char.h
hci_bcsp.c  hci_ll.c     hci_wrapper.c  Makefile    modules.order
hci_h4.c    hci_ps.c     hci_wrapper.h  mbt_char.c  README
$> make clean && make build
find . -name "*.o" -exec rm {} \;
find . -name "*.ko" -exec rm {} \;
find . -name ".*.cmd" -exec rm {} \;
find . -name "*.mod.c" -exec rm {} \;
find . -name "*.symvers" -exec rm {} \;
rm -rf .tmp_versions
make -C …
make[1]: Entering directory …
…
$> ls
bt_drv.h    hci_ldisc.c  hci_ps.o        hci_uart.o     Makefile
hci_bcsp.c  hci_ldisc.o  hci_uart.h      hci_wrapper.c  mbt_char.c
hci_bcsp.o  hci_ll.c     hci_uart.ko     hci_wrapper.h  mbt_char.h
hci_h4.c    hci_ll.o     hci_uart.mod.c  hci_wrapper.o  mbt_char.o
hci_h4.o    hci_ps.c     hci_uart.mod.o  include        modules.order
README      Module.sysmvers
$> cd ${TOP}/p195/SD-UAPSTA-BT-UART-8977-U16-MMC-W16.68.1.p195-
16.26.1.p195-C4X16625_V2-MGPL
```

```
$> ls
bin_muart  muart_src  wlan_src
```

➔ An additional subdirectory has been created. The driver and utility binaries can be found in the directory `.../bin_muart`.

5.4 Compiling WLAN Driver Sources

5.4.1 Building Result of Driver Sources

Following Kernel modules and utilities (configuration tools) will be generated or copied by compiling the WLAN driver sources:

File Name	Description
mlan.ko	Kernel driver for event handling through the netlink layer and ioctl calls
sd8977.ko	WLAN Device Kernel driver (mlan0, uap0, wfd0, nan0 device)
mlanutl	Utility for configuration additional parameters available for mdriver
mlanevent.exe	Utility to listen for and obtain events from the driver through the netlink layer
mlan2040coex	Application handles the 802.11n 20/40 coexistence operation for mdriver
uaputl.exe	Utility to get and set uAP's settings
load	WLAN driver load shell script
unload	WLAN driver unload shell script
README*	Readme files as user manual for the utilities with instructions and examples
config/*	Subdirectory for various sample configuration files used by utilities

5.4.2 Compiling WLAN Driver

1. Change directory to the WLAN driver source subdirectory:

```
$> cd ${TOP}/p195/SD-UAPSTA-BT-8977-U16-MMC-W16.68.1.p195-
16.26.1.p195-C4X16625_V2-MGPL
$> ls
bin_sd8977_bt  bin_sd8977_btchar  mbtc_src  mbt_src  wlan_src
$> cd wlan_src
$> ls
gpl-2.0.txt  mapp  mlinux  README_MLAN     README_UAP          script
Makefile     mlan  README  README_OPENWRT  README_WIFIDIRECT
$> make clean && make build
find . -name "*.o" -exec rm {} \;
find . -name "*.ko" -exec rm {} \;
find . -name ".*.cmd" -exec rm {} \;
find . -name "*.mod.c" -exec rm {} \;
find . -name "Module.symvers" -exec rm {} \;
```

```
find . -name "Module.markers" -exec rm {} \;
find . -name "modules.order" -exec rm {} \;
find . -name ".*.dwo" -exec rm {} \;
rm -rf .tmp_versions
make -C mapp/mlanconfig clean
…
cp -rpf script/wifidirect ../bin_sd8977
cp -rpf script/wifidisplay ../bin_sd8977
make -C mapp/wifidirectutl build INSTALLDIR=../bin_sd8977
…
make -C mapp/mlanevent build INSTALLDIR=../bin_sd8977
make[1]: Leaving directory …
$> ls
gpl-2.0.txt  mlan   mlan.mod.o  modules.order   README_MLAN
README_WIFIDIRECT  sd8xxx.mod.c Makefile  mlan.ko  mlan.o
Module.symvers  README_OPENWRT  script  sd8xxx.mod.o  mapp
mlan.mod.c  mlinux  README  README_UAP  sd8xxx.ko  sd8xxx.o
$> cd ${TOP}/p195/SD-UAPSTA-BT-8977-U16-MMC-W16.68.1.p195-
16.26.1.p195-C4X16625_V2-MGPL
$> ls
bin_sd8977  bin_sd8977_bt  bin_sd8977_btchar  mbtc_src  mbt_src
wlan_src
```

➔ An additional subdirectory has been created. The driver and utility binaries can be found in the directory `.../bin_sd8xxx`.

6 WLAN Device Driver

6.1 Install WLAN Device Driver Kernel and Firmware

All available WLAN driver options of Marvell® device Kernel module are listed in the file `README` which is included in the software package.

1. Change directory to the compiled WLAN driver binary directory
2. Use the Linux command `insmod` or `modprobe` to install the Kernel driver

```
$> cd ${TOP}/p195/SD-UAPSTA-BT-8977-U16-MMC-W16.68.1.p195-
16.26.1.p195-C4X16625_V2-MGPL/bin_sd8977
$> ls
config          mlan.ko       README_UAP          unload
load            mlanutl       README_WIFIDIRECT   wifidirect
mlan2040coex    README        sd8977.ko           wifidirectutl
mlanevent.exe   README_MLAN   uaputl.exe          wifidisplay
$> insmod mlan.ko
$> insmod sd8977.ko drv_mode=3 fw_name=mrvl/sdsd8977_combo_v2.bin
   cal_data_cfg=none cfg80211_wext=0xf cfg80211_drcs=0
```

Important Kernel Driver Options

Calibration Data Configuration Parameter (cal_data_cfg)

To get use of the calibration parameter, which are pre-stored on the one-time-programmable memory of the PAN9026, the following driver option must be set:

```
cal_data_cfg=none
```

Dual-Rapid-Channel-Selection Parameter (cfg80211_drcs)

If the simultaneous and independent WLAN and Bluetooth operation is configured, it is necessary to disable the Dual-Rapid-Channel-Selection (DRCS) mechanism:

```
cfg80211_drcs=0
```

By enabling the Dual-Rapid-Channel-Selection mechanism it is not possible to use the Bluetooth interface, but it is possible to generate a second micro-AP by setting the max_uap_bss parameter:

```
cfg80211_drcs=1 max_uap_bss=2
```

Kernel Driver Configuration

WLAN Driver Mode Option (drv_mode)

The parameter `drv_mode` should be configured by following bit settings:

- Bit 0: STA
- Bit 1: uAP
- Bit 2: WIFIDIRECT
- Bit 4: NAN

To load driver in STA only mode, use: **drv_mode=1**.

To load driver in UAP only mode, use: **drv_mode=2**.

To load driver in STA+UAP mode, use: **drv_mode=3**.

To load driver in STA+UAP+WFD+NAN mode, use: **drv_mode=23 (0x17)**.

The default mode is STA+UAP+WFD: **drv_mode=7**.

System-on-Chip Firmware Definition (fw_name)

During loading of the WLAN device Kernel module the SoC firmware is downloaded to the PAN9026 module. The SoC firmware binaries are stored at the Linux vendor specific library path `/lib/firmware/mrvl`. By using the driver option `fw_name` it is possible to define the SoC firmware binary file. To specify the WLAN/Bluetooth combo firmware use the following option:

```
fw_name=mrvl/sdsd8977_combo_v2.bin
```

To specify the WLAN firmware (without Bluetooth firmware part) for parallel download use following option:

```
fw_name=mrvl/sd8977_wlan_v2.bin fw_serial=0
```

In case of using the WLAN firmware with parallel download option the Bluetooth firmware can be downloaded by installing the Bluetooth Kernel module.

WLAN Network Link Configuration (cfg80211_wext)

The WLAN device Kernel module is linked through the Linux “cfg80211” Kernel module and the “nl80211” network link layer to the Linux user space. In parallel to the “nl80211” based user space applications like `wpa_supplicant` or `hostapd` it is possible to use the Wireless Extensions (WExt) toolchain (e.g. iwconfig, iwlist, etc.). Please configure the following driver parameter:

```
cfg80211_wext=0xf
```

WLAN Enhanced Power Management Features (auto_ds, ps_mode)

The power management features of WLAN driver are set by default to MLAN firmware default (`auto_ds=0`, `ps_mode=0`). For particular WLAN use cases it is recommended and sometimes necessary to disable power management features like WLAN deep sleep or 802.11 PSM mode as time critical events from host driver need to wake up the WLAN device in time. Following the power management options of driver which can be disabled with driver loading:

```
auto_ds=2 ps_mode=2
```

Implementation of RF Design related Power Limits (txpwrlimit_cfg)

During loading of the WLAN device Kernel module the SoC firmware is downloaded to the PAN9026 module. By using the driver option “txpwrlimit_cfg” it is possible to define a power limit table for the modules RF design related parameter. The txpower configuration file is stored at the Linux vendor specific library path `/lib/firmware/mrvl`. To specify the txpower configuration file use the following option:

```
txpwrlimit_cfg=txpowerlimit_cfg.conf
```

It is also possible to use a specific country domain related TX power table configuration file in binary format, which must be available in the Linux vendor specific library path `/lib/firmware/mrvl`. This TX power table should include also the modules RF design related parameter:

```
txpwrlimit_cfg=txpower_${COUNTRY}.bin
```

Definition of Variable

`COUNTRY` = ISO 3166-1 alpha-2 code (See “Full list of country codes” https://www.iso.org/publication/PUB500001.html)

Example: `txpower_US.bin` for `COUNTRY` = US

Implementation of Power Limits for various Domains (cntry_txpwr)

By using the driver option `txpwrlimit_cfg` it is possible to define a power limit table for the modules RF design related parameter. Additionally it is possible to apply power tables during runtime of WLAN device for restricted power limits for a specific regulatory domain (based on country code setting in the CRDA system). The TX power configuration binaries are stored at the Linux vendor specific library path `/lib/firmware/mrvl` and will be loaded dynamically if Country Code is changed by CRDA. Each country domain needs its own binary file. To specify the TX power configuration file and the dynamic change use the following options:

```
txpwrlimit_cfg=txpower_${COUNTRY}.bin cntry_txpwr=1
```

The RF design related power limits and restricted power limits for a specific regulatory domain are pre-stored on the OTP memory of regulatory domain specific PAN9026 models (US, EU, CA, MR). For these PAN9026 variants it is not necessary to load the power configuration file. For the PAN9026 world-wide model it is necessary to load the power configuration file.

For more information please refer to the Panasonic website for related documents ⇨ 10.2 Product Information.

6.2 WLAN Device Driver Loading

The firmware image is downloaded to the module if the bus driver detects the module's SDIO interface. Following the Kernel messages after the hardware interface has been detected:

```
kernel: mmc0: new high speed SDIO card at address 0001
kernel: wlan: Loading MWLAN driver
kernel: vendor=0x02DF device=0x9145 class=0 function=1
kernel: SDIO: max_segs=128 max_seg_size=65535
kernel: rx_work=1 cpu_num=2
kernel: wlan: Enable TX SG mode
kernel: wlan: Enable RX SG mode
kernel: Request firmware: sdsd8977_combo_v2.bin
kernel: Wlan: FW download over, firmwarelen=587052 downloaded 587052
kernel: WLAN FW is active
kernel: fw_cap_info=0x18fccf03, dev_cap_mask=0xffffffff
kernel: max_p2p_conn = 8, max_sta_conn = 8
kernel: ForceRegionRule is set in the on-chip OTP memory
kernel: 11D: Error setting domain info in FW
kernel: get_channel when STA is not connected
kernel: get_channel when AP is not started
kernel: wlan: version = SD8977-16.68.1.p195-C4X16C625-GPL-(FP68)
kernel: wlan: Driver loaded successfully
```

After hardware initialization the firmware is running on the module and network devices are registered on the Linux host system:

Interface	Description
mlan0	WLAN Station / Client mode
uap0	WLAN micro access point mode (uAP)
wfd0	Wi-Fi Direct™ mode (WFD Client or Group Owner)
nan0	Neighbor Awareness Networking mode (Wi-Fi Aware™)

After complete loading of the WLAN Kernel driver and the hardware is initialized following Kernel modules are listed at the system:

```
$> lsmod
Module                  Size      Used by
sd8xxx                450560      0
mlan                  438272      1 sd8xxx
```

6.3 Remove WLAN Device Driver

Before the WLAN Kernel driver can be removed it is necessary to shut down the Ethernet devices, which have been configured by WLAN driver mode option `drv_mode` and brought-up for network-manager service.

Driver mode STA: `drv_mode=[1,3,5,7]` → `ifconfig mlan0 down`
Driver Mode UAP: `drv_mode=[2,3,6,7]` → `ifconfig uap0 down`
Driver Mode WFD: `drv_mode=[4,5,6,7]` → `ifconfig wfd0 down`
Driver Mode NAN: `drv_mode=[16 … 23]` → `ifconfig nan0 down`

The following requisites must be met:

✓ Ethernet devices are shut down

The loaded WLAN device driver can be removed as following:

```
$> ifconfig mlan0 down
$> ifconfig uap0 down
$> ifconfig wfd0 down
$> ifconfig nan0 down
$> rmmod sd8xxx
$> rmmod mlan
```

7 Bluetooth HCI Device Driver

7.1 Install Bluetooth HCI Device Driver (SD-BT)

All available Bluetooth driver options of Marvell® device Kernel module are listed in the file `README` which is included in the software package.

1. Change directory to the compiled Bluetooth driver binary directory
2. Use the Linux command `insmod` to install the Marvell Kernel driver:

```
$> cd ${TOP}/p195/SD-UAPSTA-BT-8977-U16-MMC-W16.68.1.p195-
16.26.1.p195-C4X16625_V2-MGPL/bin_sd8977_bt
$> ls
bt8977.ko  config  fmapp  load  README  unload
$> insmod bt8977.ko
```

Important Kernel Driver Options

During loading of the WLAN device Kernel module usually the WLAN/Bluetooth combo firmware is uploaded to the PAN9026 module. The firmware binaries are stored at the Linux vendor specific library path `/lib/firmware/mrvl`. By using the driver option `fw_name` it is possible to define only the Bluetooth firmware binary file. To specify only the Bluetooth firmware for parallel download use following option:

```
fw_name=mrvl/sd8977_bt_v2.bin  bt_fw_serial=0
```

In case of using the WLAN firmware with parallel download option the Bluetooth firmware can be downloaded by installing the Bluetooth Kernel module.

7.2 Install Bluetooth HCI Device Driver (UART-BT)

All available Bluetooth driver options of Marvell® device Kernel module are listed in the file `README` which is included in the software package.

1. Change directory to the compiled Bluetooth driver binary directory.
2. Use the Linux command `insmod` to install the Marvell Kernel driver.

```
$> cd ${TOP}/p195/SD-UAPSTA-UART-BT-8977-U16-MMC-W16.68.1.p195-
16.26.1.p195-C4X16625_V2-MGPL/bin_muart
$> ls
hci_uart.ko  README
$> insmod hci_uart.ko reset=1 wakeupmode=1
```

7.3 Bluetooth Device Driver Loading

Usually the Bluetooth SoC firmware is downloaded combined with the WLAN SoC firmware. By loading of the Bluetooth Device Kernel the driver activates the Bluetooth firmware and initializes the module's hardware. In case of using the WLAN firmware with parallel download option the Bluetooth firmware is downloaded by installing the Bluetooth Kernel module. After loading of the Bluetooth Device Kernel driver following Kernel messages appear:

```
kernel: BT: Loading driver
kernel: BT Request firmware: sd8977_bt_v2.bin
kernel: BT: FW download over, size 171748 bytes
kernel: BT FW is active
kernel: BT: Driver loaded successfully
```

A new Bluetooth device is registered on the Linux host system:

Interface	Description
hci0	Bluetooth device interface

After loading of the Bluetooth driver the following Kernel module is listed at the system:

```
$> lsmod
Module                  Size      Used by
bt8xxx                 81920      1
```

7.4 Bluetooth Stack and Services

The Bluetooth Device Kernel driver is interconnected to the BlueZ stack.

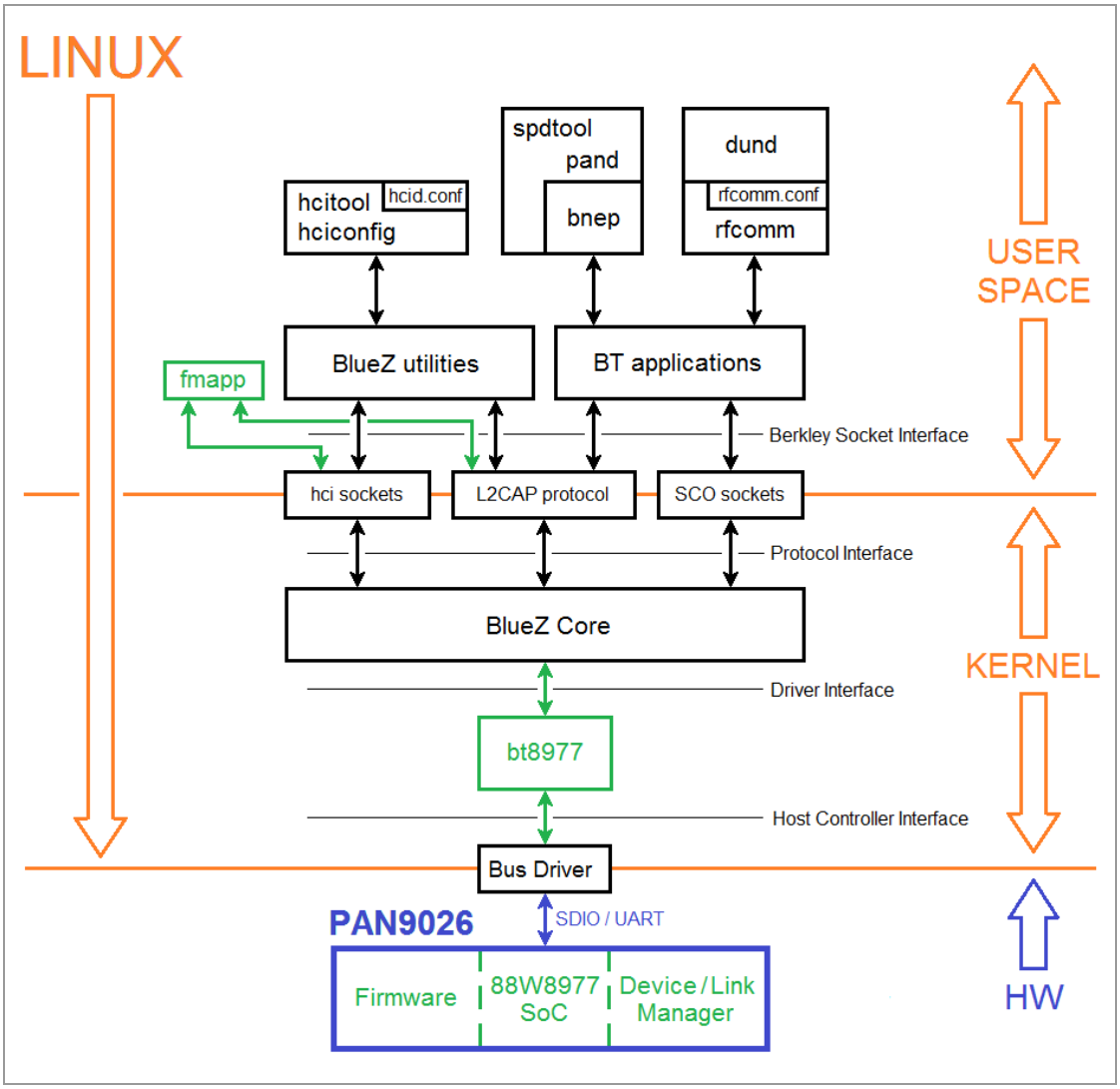
The Bluetooth "bt8977" Device Kernel requires the installation of the BlueZ stack on the Linux system, which supports the core Bluetooth layers and protocols. Additionally applications like Bluetooth service (bluetoothd), rfkill, and rfcomm can be installed.

Installation of software packages for enabling the Bluetooth service (using “system” daemon):

```
$> sudo apt-get install bluetooth bluez bluez-tools rfkill rfcomm
$> sudo systemctl start bluetooth.service
$> sudo systemctl enable bluetooth.service
```

7.5 Bluetooth Protocol Stack

The Linux system provides a standard Bluetooth protocol stack by BlueZ. The Bluetooth Device Kernel driver runs on top of the SDIO bus driver. The driver forwards data and commands between the Bluetooth SoC firmware and the driver interface of BlueZ core. Additionally private commands are handled from the Bluetooth driver. By loading of the Bluetooth Device Kernel driver the hardware is registered by the bus driver, the Bluetooth SoC firmware is initialized, and an HCI device is registered with the BlueZ stack.



7.6 Bluetooth HCI Device Interface (SD-BT)

The Bluetooth HCI Device can be controlled by Linux Bluetooth Services:

After having installed the Bluetooth “bt8977” Device Kernel, the Bluetooth service is creating an interface with the name “hci0”. The tool “hciconfig” can be used to bring-up the Bluetooth interface.

Bringing up the Bluetooth HCI Device with the following command:

```
$> hciconfig hci0 up
```

Blocked Bluetooth Device

If the interface “hci0” cannot be brought-up make sure that the Bluetooth device is turned on and not blocked.

Unblock Bluetooth HCI Device with the following command:

```
$> sudo rfkill unblock bluetooth
$> hciconfig hci0 up
```

7.7 Bluetooth HCI Device Interface (UART-BT)

The Bluetooth HCI Device can be controlled by Linux Bluetooth Services:

After having installed the Bluetooth “bt8977” Device Kernel, the HCI Device connected via UART interface to the Linux Host system need to be attached for interconnecting to BlueZ stack. The tool “hciconfig” can be used to bring-up the Bluetooth device.

Attach serial device via UART HCI to BlueZ stack with the following command:

```
$> hciattach -b /dev/ttyS1 any 3000000 flow
```

Bringing up the Bluetooth HCI Device with the following command:

```
$> hciconfig hci0 up
```

7.8 Control Bluetooth HCI Device

hcitool

The Bluetooth device can be configured and controlled by the Linux “hcitool”. The “hcitool” is used to configure Bluetooth connections and send special command to Bluetooth devices.

Usage of hcitool

```
$> sudo hcitool -i hci0 scan
Scanning ...
       00:13:43:75:91:80      pan9026_bt
$> sudo l2ping -i hci0 00:13:43:75:91:80
Ping: 00:13:43:75:91:80 from 00:13:43:88:91:38 (data size 44)
4 bytes from 00:13:43:75:91:80 id 0 time 15.38ms
4 bytes from 00:13:43:75:91:80 id 1 time 6.17ms
```

Setting example of Bluetooth Device by hcitool

```
$> sudo hcitool -i hci0 cmd 0x03 0x0003
$> sudo hcitool -i hci0 cmd 0x03 0x001a 0x03
$> sudo hcitool -i hci0 cmd 0x03 0x0005 0x02 0x00 0x03
$> sudo hcitool -i hci0 cmd 0x06 0x0003
```

Description of Settings

Bluetooth HCI Device `hci0` settings:

- Reset
- Write scan enable
- Set event filter to allow all connections with role “switchb”
- Enable device under test mode

7.9 Remove Bluetooth HCI Device

Removing the Bluetooth HCI Device (SD-BT) with the following command:

```
$> hciconfig hci0 down
$> rmmod bt8xxx
```

Removing the Bluetooth HCI Device (UART-BT) with the following command:

```
$> hciconfig hci0 down
$> sudo killall -9 hciattach
$> rmmod hci_uart
```

8 Bluetooth Character Device Driver

8.1 Install Bluetooth Character Device Driver

1. Change directory to the compiled Bluetooth driver binary directory with the following command.
2. Use the Linux command `insmod` to install the Marvell Kernel driver.

```
$> cd ${TOP}/p195/SD-UAPSTA-BT-8977-U16-MMC-W16.68.1.p195-
16.26.1.p195-C4X16625_V2-MGPL/bin_sd8977_btchar
$> ls
mbt8977.ko  config  fmapp  load  README  unload
$> insmod mbt8977.ko
```

All available Bluetooth driver options of Marvell device Kernel module are listed in the file `README` which is included in the software package.

Important Kernel Driver Options

During loading of the WLAN device Kernel module usually the WLAN/Bluetooth combo firmware is uploaded to the PAN9026 module. The firmware binaries are stored at the Linux vendor specific library path `/lib/firmware/mrvl`. By using the driver option `fw_name` it is possible to define only the Bluetooth firmware binary file. To specify only the Bluetooth firmware for parallel download use following option:

```
fw_name=mrvl/sd8977_bt_v2.bin bt_fw_serial=0
```

In case of using the Bluetooth firmware with parallel download option the WLAN firmware can be downloaded by installing the WLAN Kernel module.

8.2 Bluetooth Character Device Interface

The Bluetooth Character Device can be controlled through following interface: The Bluetooth “mbt8977” Device Kernel is creating a device interface at the Linux device directory `/dev/mbtchar0`.

8.3 Control Bluetooth Character Device

The Bluetooth Character Device can be controlled by a Marvell utility. After having installed the Bluetooth “mbt8977” Device Kernel, the Kernel driver is creating an interface with name “mbtchar0”. The utility “fmapp” can be used to control the Bluetooth character interface.

Setting example of Bluetooth Character Device by utility "fmapp":

```
$> ./fmapp mbtchar0 cmd 0x03 0x0003
$> ./fmapp mbtchar0 cmd 0x03 0x001a 0x03
$> ./fmapp mbtchar0 cmd 0x03 0x0005 0x02 0x00 0x03
$> ./fmapp mbtchar0 cmd 0x06 0x0003
```

Description of Settings

Bluetooth Character Device `mbtchar0` setting:

- Reset
- Write scan enable
- Set event filter to allow all connections with role "switch"
- Enable device under test mode

8.4 Remove Bluetooth Character Device

The Bluetooth HCI Device can be removed with the following command:

```
$> rmmod mbt8xxx
```

9 WLAN Utilities

Marvell® provides utilities for the user space to configure, control, and obtain events for driving the WLAN device in various modes.

9.1 MLAN Utility for Station Mode

Execute the MLANUTL utility in the subdirectory of WLAN driver and utility binaries with the following command:

```
$> cd ${TOP}/p195/SD-UAPSTA-BT-8977-U16-MMC-W16.68.1.p195-
16.26.1.p195-C4X16625_V2-MGPL/bin_sd8977
```

9.1.1 Enabling Clear Channel Assessment (CCA)

The CCA for adaptivity capability of station can be configured with the following command:

```
$> ./mlanutl -i mlan0 hostcmd config/ed_mac_ctrl_V3_8977.conf
ed_mac_ctrl_v3
```

Adaptivity Capability

The CCA is required to perform the adaptivity test in compliance with ETSI EN 300 328 V2.1.1 and ETSI EN 301 893 V2.1.1.

9.1.2 Network Interface Configuration of Station Mode

The network interface is configured by Linux Network tools with the following command:

```
$> kill -9 wpa_supplicant
$> wpa_supplicant -i mlan0 -c ${CONFIG_DIRECTORY}/${SSID}.conf -f
/tmp/wpa_supplicant.log
$> dhclient -d -v mlan0 &> /tmp/dhclient.log
```

Definition of wpa_supplicant configuration file

`SSID` = Service Set Identifier of Access Point (AP)

Content of SSID.conf:

```
network={
      ssid="SSID"
      psk="pre-shared-key"
}
```

9.1.3 Configuration of Stations TX Data Rate (txratecfg)

The TX Data Rate of Station can be configured with the following command:

```
$> ./mlanutl mlan0 txratecfg ${HTCFG} ${TXRATECFG}
```

Definition of Variables

`TXRATECFG` = [TXRATE]

`HTCFG` = [0, 1]

With `HTCFG`=0 (Legacy Mode 802.11a/b/g):

- TXRATE=0 for 1 Mbps (802.11b)
- TXRATE=1 for 2 Mbps (802.11b)
- TXRATE=2 for 5.5 Mbps (802.11b)
- TXRATE=3 for 11 Mbps (802.11b)
- TXRATE=4 for 6 Mbps (802.11a/g)
- TXRATE=5 for 9 Mbps (802.11a/g)
- TXRATE=6 for 12 Mbps (802.11a/g)
- TXRATE=7 for 18 Mbps (802.11a/g)
- TXRATE=8 for 24 Mbps (802.11a/g)
- TXRATE=9 for 36 Mbps (802.11a/g)
- TXRATE=10 for 48 Mbps (802.11a/g)
- TXRATE=11 for 54 Mbps (802.11a/g)

With `HTCFG`=1 (HT Mode 802.11n):

- TXRATE=0 for MCS0 (802.11n)
- TXRATE=1 for MCS1 (802.11n)
- TXRATE=2 for MCS2 (802.11n)
- TXRATE=3 for MCS3 (802.11n)
- TXRATE=4 for MCS4 (802.11n)
- TXRATE=5 for MCS5 (802.11n)
- TXRATE=6 for MCS6 (802.11n)
- TXRATE=7 for MCS7 (802.11n)
- TXRATE=32 for MCS32 (802.11n)

With `HTCFG`="" (empty):

TXRATE=0xff for auto rating (default)

9.1.4 Set Station for 2.4 GHz Band in Legacy Mode (802.11b/g)

The Station device can be configured with the following command:

```
$> ./mlanutl mlan0 httxcfg 0x2C 1
$> ./mlanutl mlan0 htcapinfo 0x4800000 1
```

9.1.5 Set Station for 5 GHz Band in Legacy Mode (802.11a)

The Station device can be configured with the following command:

```
$> ./mlanutl mlan0 httxcfg 0x6E 2
$> ./mlanutl mlan0 htcapinfo 0x5820000 2
```

9.1.6 Set Station for HT20 Mode (802.11n)

The Station device can be configured with the following command:

```
$> ./mlanutl mlan0 httxcfg 0x3C 0
$> ./mlanutl mlan0 htcapinfo 0x20800000 0
```

9.1.7 Set Station for HT40 Mode (802.11n)

The Station device can be configured with the following command:

```
$> ./mlanutl mlan0 httxcfg 0x5E 0
$> ./mlanutl mlan0 htcapinfo 0x21000000 0
```

9.1.8 Set Station with HT20/40 Coexistence Mechanism

The Station device can be configured with the following command:

```
$> ./mlanutl mlan0 hostcmd config/11n_2040coex.conf 2040coex
```

The HT20/40 coexistence configuration should be used in country domains, where 802.11n HT20 WLAN channel and HT40 WLAN channel coexists and can overlap. By this command the firmware sends 802.11n 20/40 coexistence management frames to the external access point.

The configuration parameters are set in the file `11n_2040coex.conf` available in the driver directory `config`.

9.2 UAP Utility for Micro-AP Mode

Execute the UAPUTL utility in the subdirectory of WLAN driver and utility binaries with the following command:

```
$> cd ${TOP}/p195/SD-UAPSTA-BT-8977-U16-MMC-W16.68.1.p195-
16.26.1.p195-C4X16625_V2-MGPL/bin_sd8977
```

9.2.1 Enabling Clear-Channel-Assessment (CCA)

The CCA for adaptivity capability of BSS (Basic Service Set) can be configured with the following command:

```
$> ./uaputl.exe -i uap0 hostcmd config/ed_mac_ctrl_V3_8977.conf
ed_mac_ctrl_v3
```

Adaptivity Capability

The CCA is required to perform the adaptivity test in compliance with ETSI EN 300 328 V2.1.1 and ETSI EN 301 893 V2.1.1.

9.2.2 Stop Basic Service Set (bss_stop)

The BSS can be stopped with the following command:

```
$> ./uaputl.exe -i uap0 bss_stop
```

9.2.3 System Reset Basic Service Set (sys_reset)

The system of BSS can be reset with the following command:

```
$> ./uaputl.exe -i uap0 sys_reset
```

9.2.4 Start Basic Service Set at 2.4 GHz Band in LT Mode (802.11b)

The BSS can be configured and started with the following command:

```
$> ./uaputl.exe -i uap0 bss_stop
$> ./uaputl.exe -i uap0 sys_reset
$> ./uaputl.exe -i uap0 sys_cfg_2040_coex 1
$> ./uaputl.exe -i uap0 sys_cfg_11n 0
$> ./uaputl.exe -i uap0 sys_cfg_rates 0x82 0x84 0x8B 0x96
$> ./uaputl.exe -i uap0 sys_cfg_channel_ext ${CHANNEL} 0
$> ./uaputl.exe -i uap0 sys_cfg_ssid "PAN9026_uAP0"
$> ./uaputl.exe -i uap0 bss_start
$> ./uaputl.exe -i uap0 txratecfg 0 ${TXRATECFG}
```

Definition of Variables

CHANNEL = [1, 2, 3, ..., 11, (12), (13)]

TXRATECFG = [0, 1, 2, 3]

9.2.5 Start Basic Service Set at 2.4 GHz Band in LT Mode (802.11bg)

The BSS can be configured and started with the following command:

```
$> ./uaputl.exe -i uap0 bss_stop
$> ./uaputl.exe -i uap0 sys_reset
$> ./uaputl.exe -i uap0 sys_cfg_2040_coex 1
$> ./uaputl.exe -i uap0 sys_cfg_11n 0
$> ./uaputl.exe -i uap0 sys_cfg_rates 0x82 0x84 0x8B 0x96 0x0C 0x12
0x18 0x24 0x30 0x48 0x60 0x6C
$> ./uaputl.exe -i uap0 sys_cfg_channel_ext ${CHANNEL} 0
$> ./uaputl.exe -i uap0 sys_cfg_ssid "PAN9026_uAP0"
$> ./uaputl.exe -i uap0 bss_start
```

Definition of Variables

CHANNEL = [1, 2, 3, ..., 11, (12), (13)]

9.2.6 Start Basic Service Set at 2.4 GHz Band in HT20 Mode (802.11n)

The BSS can be configured and started with the following command:

```
$> ./uaputl.exe -i uap0 bss_stop
$> ./uaputl.exe -i uap0 sys_reset
$> ./uaputl.exe -i uap0 sys_cfg_2040_coex 1
$> ./uaputl.exe -i uap0 httxcfg 0x20
$> ./uaputl.exe -i uap0 sys_cfg_11n 1 0x012C 3
$> ./uaputl.exe -i uap0 sys_cfg_rates 0x82 0x84 0x8B 0x96 0x0C 0x12
0x18 0x24 0x30 0x48 0x60 0x6C
$> ./uaputl.exe -i uap0 sys_cfg_channel_ext ${CHANNEL} 0
$> ./uaputl.exe -i uap0 sys_cfg_ssid "PAN9026_uAP0"
$> ./uaputl.exe -i uap0 bss_start
```

Definition of Variables

CHANNEL = [1, 2, 3, ..., 11, (12), (13)]

9.2.7 Start Basic Service Set at 2.4 GHz Band in HT40 Mode (802.11n)

The BSS can be configured and started with the following command:

```
$> ./uaputl.exe -i uap0 bss_stop
$> ./uaputl.exe -i uap0 sys_reset
$> ./uaputl.exe -i uap0 sys_cfg_2040_coex 0
$> ./uaputl.exe -i uap0 httxcfg 0x62
$> ./uaputl.exe -i uap0 sys_cfg_11n 1 0x016E 3
$> ./uaputl.exe -i uap0 sys_cfg_rates 0x82 0x84 0x8B 0x96 0x0C 0x12
0x18 0x24 0x30 0x48 0x60 0x6C
$> ./uaputl.exe -i uap0 sys_cfg_channel_ext ${CHANNEL} 1
${SECONDARY_CHANNEL}
$> ./uaputl.exe -i uap0 sys_cfg_ssid "PAN9026_uAP0"
$> ./uaputl.exe -i uap0 bss_start
```

Definition of Variables

CENTER_CHANNEL = [3, 6, (7), 9, (11)]

SECONDARY_CHANNEL = [2, 4]

- with SECONDARY_CHANNEL=2 (above)
  CHANNEL=${CENTER_CHANNEL - 2}
- with SECONDARY_CHANNEL=4 (below)
  CHANNEL=${CENTER_CHANNEL + 2}

9.2.8 Start Basic Service Set at 5 GHz Band in LT Mode (802.11a)

The BSS can be configured and started with the following command:

```
$> ./uaputl.exe -i uap0 bss_stop
$> ./uaputl.exe -i uap0 sys_reset
$> ./uaputl.exe -i uap0 sys_cfg_2040_coex 1
$> ./uaputl.exe -i uap0 sys_cfg_11n 0
$> ./uaputl.exe -i uap0 sys_cfg_rates 0x8C 0x12 0x98 0x24 0xB0 0x48
0x60 0x6C
$> ./uaputl.exe -i uap0 sys_cfg_channel_ext ${CHANNEL} 1
$> ./uaputl.exe -i uap0 sys_cfg_ssid "PAN9026_uAP0"
$> ./uaputl.exe -i uap0 bss_start
```

Definition of Variables

CHANNEL = [36, 40, 44, 48, 52, 56, 60, 64, 100, 104, …, 165]

9.2.9 Start Basic Service Set at 5 GHz Band in HT20 Mode (802.11n)

The BSS can be configured and started with the following command:

```
$> ./uaputl.exe -i uap0 bss_stop
$> ./uaputl.exe -i uap0 sys_reset
$> ./uaputl.exe -i uap0 sys_cfg_2040_coex 1
$> ./uaputl.exe -i uap0 httxcfg 0x20
$> ./uaputl.exe -i uap0 sys_cfg_11n 1 0x012C 3
$> ./uaputl.exe -i uap0 sys_cfg_rates 0x8C 0x12 0x98 0x24 0xB0 0x48
0x60 0x6C
$> ./uaputl.exe -i uap0 sys_cfg_channel_ext ${CHANNEL} 1
$> ./uaputl.exe -i uap0 sys_cfg_ssid "PAN9026_uAP0"
$> ./uaputl.exe -i uap0 bss_start
```

Definition of Variables

CHANNEL = [36, 40, 44, 48, 52, 56, 60, 64, 100, 104, …, 165]

9.2.10 Start Basic Service Set at 5 GHz band in HT40 Mode (802.11n)

The BSS can be configured and started with the following command:

```
$> ./uaputl.exe -i uap0 bss_stop
$> ./uaputl.exe -i uap0 sys_reset
$> ./uaputl.exe -i uap0 sys_cfg_2040_coex 0
$> ./uaputl.exe -i uap0 httxcfg 0x62
$> ./uaputl.exe -i uap0 sys_cfg_11n 1 0x016E 3
$> ./uaputl.exe -i uap0 sys_cfg_rates 0x8C 0x12 0x98 0x24 0xB0 0x48
0x60 0x6C
$> ./uaputl.exe -i uap0 sys_cfg_channel_ext ${CHANNEL} 1
${SECONDARY_CHANNEL}
$> ./uaputl.exe -i uap0 sys_cfg_ssid "PAN9026_uAP0"
$> ./uaputl.exe -i uap0 bss_start
```

Definition of Variables

`CENTER_CHANNEL` = [38, 46, 54, 62, 102, 110, 118, …, 159]

`SECONDARY_CHANNEL` = [2 , 4]

- with `SECONDARY_CHANNEL=2` (above)
  `CHANNEL=${CENTER_CHANNEL - 2}`
- with `SECONDARY_CHANNEL=4` (below)
  `CHANNEL=${CENTER_CHANNEL + 2}`

9.2.11 Configuration of TX Data Rate for BSS (txratecfg)

The TX Data Rate of BSS can be configured with the following command:

```
$> ./uaputl.exe -i uap0 txratecfg ${HTCFG} ${TXRATECFG}
```

Definition of Variables

`TXRATECFG` = [TXRATE]

`HTCFG` = [0, 1]

With `HTCFG`=0 (Legacy Mode 802.11a/b/g)

- TXRATE=0 for 1 Mbps (802.11b)
- TXRATE=1 for 2 Mbps (802.11b)
- TXRATE=2 for 5.5 Mbps (802.11b)
- TXRATE=3 for 11 Mbps (802.11b)
- TXRATE=4 for 6 Mbps (802.11a/g)
- TXRATE=5 for 9 Mbps (802.11a/g)
- TXRATE=6 for 12 Mbps (802.11a/g)
- TXRATE=7 for 18 Mbps (802.11a/g)
- TXRATE=8 for 24 Mbps (802.11a/g)
- TXRATE=9 for 36 Mbps (802.11a/g)
- TXRATE=10 for 48 Mbps (802.11a/g)
- TXRATE=11 for 54 Mbps (802.11a/g)

With `HTCFG`=1 (HT Mode 802.11n)

- TXRATE=0 for MCS0 (802.11n)
- TXRATE=1 for MCS1 (802.11n)
- TXRATE=2 for MCS2 (802.11n)
- TXRATE=3 for MCS3 (802.11n)
- TXRATE=4 for MCS4 (802.11n)
- TXRATE=5 for MCS5 (802.11n)
- TXRATE=6 for MCS6 (802.11n)
- TXRATE=7 for MCS7 (802.11n)
- TXRATE=32 for MCS32 (802.11n)

With `HTCFG`="" (empty)

TXRATE=0xff for Auto Rating (default)

Changing of TX Data Rate

Changing of TX Data Rate is only possible after the BSS has been started.

9.2.12 Read System Configuration of Basic Service Set (sys_config)

The system configuration of BSS can be red with the following command:

```
$> ./uaputl.exe -i uap0 sys_config
```

9.2.13 Country Code Setting of Basic Service Set (sys_cfg_80211d)

The country code setting of BSS can be configured with the following command:

```
$> ./uaputl.exe -i uap0 sys_cfg_80211d state 1 country ${COUNTRY}
```

Definition of Variable

`COUNTRY` = ISO 3166-1 alpha-2 code (See “Full list of country codes” https://www.iso.org/publication/PUB500001.html)

Examples:

- US (United States)
- CA (Canada)
- DE (Germany)
- AU (Australia)
- NZ (New Zealand)

If the country code setting is forced in the OTP memory it is not possible to change the setting in SoC firmware by driver. For more information please refer to the Panasonic website for related documents ⇨ 10.2 Product Information.

9.2.14 Network Interface Configuration of Basic Service Set

The network interface and DHCP-Server for BSS can be configured with the following command:

```
$> ifconfig uap0 192.168.33.1 netmask 255.255.255.0
$> kill -9 dhcpd
$> dhcpd -4 -f -d -cf ${CONFIG_DIRECTORY}/dhcpd.conf uap0
```

dhcpd.conf file content

```
subnet 192.168.33.0 netmask 255.255.255.0 {
      range 192.168.33.10 192.168.33.200;
      option routers 192.168.33.1;
      option domain-name-servers 8.8.8.8;
}
```

9.3 UAP Utility Configuration File

9.3.1 Load Basic Service Set Settings from UAPUTL Configuration File

The BSS can be configured and started with the following command:

```
$> ./uaputl.exe -i uap0 bss_stop
$> ./uaputl.exe -i uap0 sys_reset
$> ./uaputl.exe -i uap0 sys_config config/uaputl.conf
$> ./uaputl.exe -i uap0 bss_config config/uaputl.conf
$> ./uaputl.exe -i uap0 bss_start
```

UAPUTL Configuration File

The file `uaputl.conf` is available in the subdirectory `config` of the WLAN driver directory `bin_sd8xxx`.

9.3.2 Default UAPUTL Configuration File

Default BSS Configuration Parameter

Configuration Parameter ap_config={ … }	Parameterize	Remark
SSID	Marvell Micro AP	SSID of Micro AP
BeaconPeriod	100	Beacon period in TU
Channel	6	Radio channel 6 (Band=0)
ChanList	1,6,11	Scan channel list (Band=0)
Band	0	0 for 2.4 GHz band 1 for 5 GHz band
RxAntenna	0	0 for Antenna A
TxAntenna	0	1 for Antenna B
Rate	0x82,0x84,0x8b,0x96, 0x0C,0x12,0x18,0x24, 0x30,0x48,0x60,0x6c	Set of data rate that a station in the BSS use
TxPowerLevel	13	Transmit power level in dBm
BroadcastSSID	1	Broadcast SSID feature (1: Enable, 0: Disable)
RTSThreshold	2 347	RTS threshold value
FragThreshold	2 346	Fragmentation threshold value
DTIMPeriod	1	DTIM period in beacon periods
MCBCdataRate	0	MCBC rate to use for packet transmission (0: auto)
TxBeaconRate	0	Beacon rate to use for Beacon transmission (0: auto)

Configuration Parameter ap_config={ … }	Parameterize	Remark
PktFwdCtl	1	Packet forwarding control
StaAgeoutTimer	1800	Inactive client station age out timer value in units of 100 ms
PSStaAgeoutTimer	400	Inactive client PS station age out timer value in units of 100 ms
MaxStaNum	10	Max number of stations allowed to connect. Need to change to MaxStaNum=8 for “8977”.
Retrylimit	7	Retry limit to use for packet transmissions
AuthMode	0	0: Open authentication/1: shared key authentication/3: WPA3 SAE
Protocol	1	Protocol to use (1: No security/2: Static WEP/8: WPA/32: WPA2/40: WPA2 Mixed Mode /64: WPA3 SAE)
RSNReplayProtection	0	RSN replay protection (0: disabled, 1: enabled)
PairwiseUpdateTimeout	100	Pairwise Handshake update timeout: 100 ms
PairwiseHandshakeRetries	3	Pairwise Handshake retries: 3
GroupwiseUpdateTimeout	100	Groupwise Handshake update timeout: 100 ms
GroupwiseHandshakeRetries	3	Groupwise Handshake retries: 3
GroupRekeyTime	86400	Group key re-key interval, in second, 0 mean never re-key
Enable11n	1	1 to enable, 0 to disable
HTCapInfo	0x111c	HTCapInfo
AMPDU	0x03	AMPDU
HT_MCS_MAP	0x000000ff	Bit 7-0: MCS_SET_0, Bit 15-8: MCS_SET_1
Enable2040Coex	1	Enable 20/40 coex feature
11d_enable	0	0-disable 1-enable
country	US	Country information

The embedded Authenticator of UAPUTL utility does not support WPA3 SAE authentication mode. If WPA3 SAE authentication is required, please use the WPA3 SAE implementation of `hostapd` version 2.6.

9.3.3 Parameterize Maximum Station Number

Parameterize BSS Configuration at file `uaputl.conf` as following:

Configuration Parameter ap_config={ … }	Parameterize	Remark
MaxStaNum	8	Max number of stations for chipset “8977”

If using the file `UAPUTL.CONF` for BSS configuration of PAN9026 micro-AP it is necessary to parameterize the parameter `MaxStaNum` as following:

```
MaxStaNum=8
```

9.3.4 Parameterize SSID

Parameterize BSS Configuration at file `UAPUTL.CONF` as following:

Configuration Parameter ap_config={ … }	Parameterize	Remark
SSID	PAN9026_uAP0	SSID of Micro AP

9.3.5 Parameterize Authentication and Protocol for WPA2 Mixed Mode

Parameterize BSS Configuration at file `UAPUTL.CONF` as following:

Configuration Parameter ap_config={ … }	Parameterize	Remark
AuthMode	0	0: Open authentication (WPA/2)/1: shared key authentication (WEP)/3: WPA3 SAE
Protocol	40	Protocol to use (1: No security/2: Static WEP/8: WPA/32: WPA2/40: WPA2 Mixed Mode)
PwkCipherWPA	8	Pairwise cipher type (4: TKIP/8: AES CCMP/12: AES CCMP+TKIP)
PwkCipherWPA2	8	
GwkCipher	8	Group cipher type (4: TKIP/8: AES CCMP)
PSK	1234567890	WPA/WPA2 passphrase
GroupRekeyTime	86400	Group key re-key interval, in second, 0 mean never re-key

9.3.6 Parameterize Authentication and Protocol for WPA2 Mode

Parameterize BSS Configuration at file `UAPUTL.CONF` as following:

Configuration Parameter ap_config={ … }	Parameterize	Remark
AuthMode	0	0: Open authentication (WPA/2)/1: shared key authentication (WEP)/3: WPA3 SAE
Protocol	32	Protocol to use (1: No security/2: Static WEP/8: WPA/32: WPA2/40: WPA2 Mixed Mode)
PwkCipherWPA	8	Pairwise cipher type (4: TKIP/8: AES CCMP/12: AES CCMP+TKIP)
PwkCipherWPA2	8	
GwkCipher	8	Group cipher type (4: TKIP/8: AES CCMP)
PSK	1234567890	WPA/WPA2 passphrase
GroupRekeyTime	86400	Group key re-key interval, in second, 0 mean never re-key

9.3.7 Enable RSN Replay Protection for WPA2 (VU#228519)

Parameterize BSS Configuration at file `UAPUTL.CONF` as following:

Configuration Parameter ap_config={ … }	Parameterize	Remark
RSNReplayProtection	1	RSN replay protection (0: disabled, 1: enabled)
PairwiseUpdateTimeout	100	Pairwise Handshake update timeout: 100 ms
PairwiseHandshakeRetries	3	Pairwise Handshake retries: 3
GroupwiseUpdateTimeout	100	Groupwise Handshake update timeout: 100 ms
GroupwiseHandshakeRetries	3	Groupwise Handshake retries: 3

The embedded Authenticator of UAPUTL utility supports the WPA2 security improvements (“VU#228519”) for handling RSNE unexpected values, verify RSN capabilities, RSNE bounds verification for WPA2-PSK, unknown root CA detection, and replay protection.

9.3.8 Parameterize Channel in 2.4 GHz and with HT Mixed Mode (802.11bgn)

Parameterize BSS Configuration at file `uaputl.conf` as following:

Configuration Parameter ap_config={ … }	Parameterize	Remark
Channel	1	Radio Channel 1 (Band=0)
Band	0	0 for 2.4 GHz band, 1 for 5 GHz band
Rate	0x82,0x84,0x8b,0x96, 0x0c,0x12,0x18,0x24, 0x30,0x48,0x60,0x6c	Set of data rate that a station in the BSS use
Enable11n	1	1 to enable, 0 to disable
HTCapInfo	0x111c	HTCapInfo
AMPDU	0x03	AMPDU
HT_MCS_MAP	0x000000ff	Bit 7-0: MCS_SET_0, Bit 15-8: MCS_SET_1
Enable2040Coex	1	Enable 20/40 coex feature

9.3.9 Parameterize Channel in 2.4 GHz with LG Mode (802.11bg)

Parameterize BSS Configuration at file `uaputl.conf` as following:

Configuration Parameter ap_config={ … }	Parameterize	Remark
Channel	1	Radio Channel 1 (Band=0)
Band	0	0 for 2.4 GHz band, 1 for 5 GHz band
Rate	0x82,0x84,0x8b,0x96, 0x0c,0x12,0x18,0x24, 0x30,0x48,0x60,0x6c	Set of data rate that a station in the BSS use
Enable11n	0	1 to enable, 0 to disable
HTCapInfo	0x111c	HTCapInfo
AMPDU	0x03	AMPDU
HT_MCS_MAP	0x000000ff	Bit 7-0: MCS_SET_0, Bit 15-8: MCS_SET_1
Enable2040Coex	1	Enable 20/40 coex feature

9.3.10 Parameterize Channel in 2.4 GHz with HT20 Mode (802.11n)

Parameterize BSS Configuration at file `uaputl.conf` as following:

Configuration Parameter ap_config={ … }	Parameterize	Remark
Channel	1	Radio Channel 1 (Band=0)
Band	0	0 for 2.4 GHz band, 1 for 5 GHz band
Rate	0x82,0x84,0x8b,0x96, 0x0c,0x12,0x18,0x24, 0x30,0x48,0x60,0x6c	Set of data rate that a station in the BSS use
Enable11n	1	1 to enable, 0 to disable
HTCapInfo	0x012c	HTCapInfo
AMPDU	0x03	AMPDU
HT_MCS_MAP	0x000000ff	Bit 7-0: MCS_SET_0, Bit 15-8: MCS_SET_1
Enable2040Coex	1	Enable 20/40 coex feature

9.3.11 Parameterize Channel in 2.4 GHz with HT40 Mode (802.11n)

Parameterize BSS Configuration at file `uaputl.conf` as following:

Configuration Parameter ap_config={ … }	Parameterize	Remark
Channel	5,2	Primary Channel 5, secondary channel above
	(9,4)[1]	Primary Channel 9, secondary channel below
Band	0	0 for 2.4 GHz band, 1 for 5 GHz band
Rate	0x82,0x84,0x8b,0x96, 0x0c,0x12,0x18,0x24, 0x30,0x48,0x60,0x6c	Set of data rate that a station in the BSS use
Enable11n	1	1 to enable, 0 to disable
HTCapInfo	0x016e	HTCapInfo
AMPDU	0x03	AMPDU
HT_MCS_MAP	0x000000ff	Bit 7-0: MCS_SET_0, Bit 15-8: MCS_SET_1
Enable2040Coex	0	Enable 20/40 coex feature

[1] Another configuration example.

9.3.12 Parameterize Channel in 5 GHz with LG Mode (802.11a)

Parameterize BSS Configuration at file `uaputl.conf` as following:

Configuration Parameter ap_config={ … }	Parameterize	Remark
Channel	36	Radio Channel 36 (Band=1)
Band	1	0 for 2.4 GHz band, 1 for 5 GHz band
Rate	0x8c,0x12,0x98,0x24, 0xb0,0x48,0x60,0x6c	Set of data rate that a station in the BSS use
Enable11n	0	1 to enable, 0 to disable
HTCapInfo	0x111c	HTCapInfo
AMPDU	0x03	AMPDU
HT_MCS_MAP	0x000000ff	Bit 7-0: MCS_SET_0, Bit 15-8: MCS_SET_1
Enable2040Coex	1	Enable 20/40 coex feature

9.3.13 Parameterize Channel in 5 GHz with HT20 Mode (802.11n)

Parameterize BSS Configuration at file `uaputl.conf` as following:

Configuration Parameter ap_config={ … }	Parameterize	Remark
Channel	36	Radio Channel 36 (Band=1)
Band	1	0 for 2.4 GHz band, 1 for 5 GHz band
Rate	0x8c,0x12,0x98,0x24, 0xb0,0x48,0x60,0x6c	Set of data rate that a station in the BSS use
Enable11n	1	1 to enable, 0 to disable
HTCapInfo	0x012c	HTCapInfo
AMPDU	0x03	AMPDU
HT_MCS_MAP	0x000000ff	Bit 7-0: MCS_SET_0, Bit 15-8: MCS_SET_1
Enable2040Coex	1	Enable 20/40 coex feature

9.3.14 Parameterize Channel in 5 GHz with HT40 Mode (802.11n)

Parameterize BSS Configuration at file `uaputl.conf` as following:

Configuration Parameter ap_config={ … }	Parameterize	Remark
Channel	36,2	Primary channel 36, secondary channel above
	(40,4)[2]	Primary channel 40, secondary channel below
Band	1	0 for 2.4 GHz band, 1 for 5 GHz band
Rate	0x8c,0x12,0x98,0x24, 0xb0,0x48,0x60,0x6c	Set of data rate that a station in the BSS use
Enable11n	1	1 to enable, 0 to disable
HTCapInfo	0x016e	HTCapInfo
AMPDU	0x03	AMPDU
HT_MCS_MAP	0x000000ff	Bit 7-0: MCS_SET_0, Bit 15-8: MCS_SET_1
Enable2040Coex	0	Enable 20/40 coex feature

9.3.15 Parameterize Country Code (802.11d)

Parameterize BSS Configuration at file `uaputl.conf` as following:

Configuration Parameter ap_config={ … }	Parameterize	Remark
11d_enable	1	0-disable 1-enable
country	US	Country information

[2] Another configuration example.

10 Contact Details

10.1 Contact Us

Please contact your local Panasonic Sales office for details on additional product options and services:

For Panasonic Sales assistance in the **EU**, visit
https://eu.industrial.panasonic.com/about-us/contact-us
Email: wireless@eu.panasonic.com

For Panasonic Sales assistance in **North America**, visit the Panasonic website "Sales & Support" to find assistance near you at
https://na.industrial.panasonic.com/distributors

Please visit the **Panasonic Wireless Technical Forum** to submit a question at
https://forum.na.industrial.panasonic.com

10.2 Product Information

Please refer to the Panasonic Wireless Connectivity website for further information on our products and related documents:

For complete Panasonic product details in the **EU**, visit
http://pideu.panasonic.de/products/wireless-modules.html

For complete Panasonic product details in **North America**, visit
http://www.panasonic.com/rfmodules