SHORT RANGE

EVK-ELLA-W1

Evaluation kit for ELLA-W1 host-based multiradio modules with Wi-Fi and Bluetooth

User Guide

Abstract

This document describes how to set up the EVK-ELLA-W1 evaluation kit to evaluate ELLA-W1 multiradio (Wi-Fi and Bluetooth) host-based modules. It also describes how to compile the Marvell Linux reference drivers and provides some basic usage examples.

www.u-blox.com

UBX-15012877 - R04

Document Information		
Title	**EVK-ELLA-W1**	
Subtitle	Evaluation kit for ELLA-W1 host-based multiradio modules with Wi-Fi and Bluetooth	
Document type	User Guide	
Document number	UBX-15012877	
Revision, date	R04	5-Oct-2016
Document status	Early Production Information	

Document status information	
Objective Specification	Document contains target values. Revised and supplementary data will be published later.
Advance Information	Document contains data based on early testing. Revised and supplementary data will be published later.
Early Production Information	Document contains data from product verification. Revised and supplementary data may be published later.
Production Information	Document contains the final product specification.

This document applies to the following products:

Product name	Type number	Firmware version	PCN / IN
EVK-ELLA-W161	EVK-ELLA-W161-A-01	**Automotive driver:** Package: SD-UAPSTA-BT-FM-8787-FC13-MMC-14.44.35.p233-M2614525_AX-GPL Firmware version: 14.44.35.p233 Driver version: M2614525 **Industrial driver:** Package: SD-UAPSTA-BT-FM-8787-FC18-MMC-14.66.35.p57-M3X14484_AX-GPL Firmware version: 14.66.35.p57 Driver version: M3X14484	N/A
EVK-ELLA-W163	EVK-ELLA-W163-A-01	**Automotive driver:** Package: SD-UAPSTA-BT-FM-8787-FC13-MMC-14.44.35.p233-M2614525_AX-GPL Firmware version: 14.44.35.p233 Driver version: M2614525 **Industrial driver:** Package: SD-UAPSTA-BT-FM-8787-FC18-MMC-14.66.35.p57-M3X14484_AX-GPL Firmware version: 14.66.35.p57 Driver version: M3X14484	N/A

u-blox reserves all rights to this document and the information contained herein. Products, names, logos and designs described herein may in whole or in part be subject to intellectual property rights. Reproduction, use, modification or disclosure to third parties of this document or any part thereof without the express permission of u-blox is strictly prohibited.

The information contained herein is provided “as is” and u-blox assumes no liability for the use of the information. No warranty, either express or implied, is given, including but not limited, with respect to the accuracy, correctness, reliability and fitness for a particular purpose of the information. This document may be revised by u-blox at any time. For most recent documents, please visit www.u-blox.com.

Copyright © 2016, u-blox AG.

u-blox® is a registered trademark of u-blox Holding AG in the EU and other countries.

Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. PCI, PCI Express, PCIe, and PCI-SIG are trademarks or registered trademarks of PCI-SIG. All other registered trademarks or trademarks mentioned in this document are property of their respective owners.

Contents

1 Evaluation kit description

1.1 Overview

The EVK-ELLA-W1 evaluation kits provide a simple way to evaluate the ELLA-W1 host-based multiradio modules with Wi-Fi and Bluetooth [2]. The EVK-ELLA-W1 evaluation board comes with on-board antennas for Wi-Fi and Bluetooth. External antennas can optionally be connected through U.FL coaxial connectors. The evaluation kit includes a standard full-size SDIO adapter board (compatible with host sockets designed for SD memory cards) for host communication.

The main features of the EVK-ELLA-W1 evaluation kit are:

- Available for single and dual antenna variants of the ELLA-W1 module
- SDIO 2.0 device interface via SDIO adapter board for host communication
- On-board dual-band 2.4/5 GHz and 2.4 GHz chip antennas for Wi-Fi and Bluetooth
- U.FL coaxial connectors for external Wi-Fi and Bluetooth antennas, switchable by 0 Ω resistors
- All module interfaces externally available
- Multiple power supply options

Table 1 lists the different evaluation kit versions:

Evaluation kit	Description	Suitable for evaluation of
EVK-ELLA-W161	Evaluation kit for versions with 1 antenna pin (shared WI-Fi and Bluetooth antenna); uses the ELLA-W161-A module	ELLA-W131, ELLA-W131-A (single band Wi-Fi) ELLA-W161, ELLA-W161-A (dual band Wi-Fi)
EVK-ELLA-W163	Evaluation kit for versions with 2 antenna pins (separate Wi-Fi and Bluetooth antennas); uses the ELLA-W163-A module	ELLA-W133, ELLA-W133-A (single band Wi-Fi) ELLA-W163, ELLA-W163-A (dual band Wi-Fi)

Table 1: List of available EVK-ELLA-W1 evaluation kits

Figure 1 shows the EVK-ELLA-W1 with the evaluation board (EVB) EVB-ELLA-W1 and the SDIO adapter.

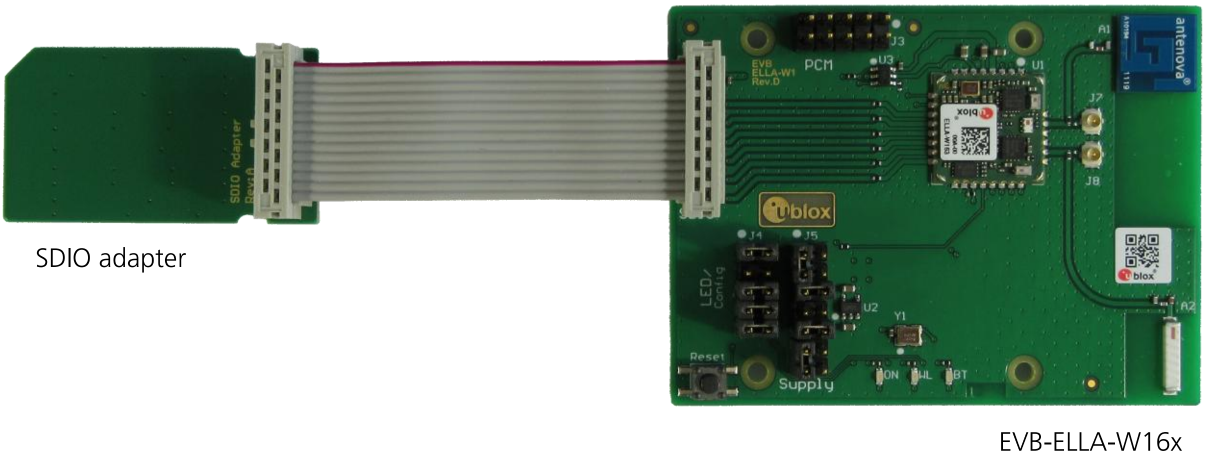

Figure 1: EVK-ELLA-W1 evaluation kit with the evaluation board[1] and SDIO adapter

[1] Figure 1 shows a preliminary version of the evaluation board.

1.2 Kit includes

The EVK-ELLA-W1 evaluation kit includes the following:

- Evaluation board EVB-ELLA-W161 or EVB-ELLA-W163
- SDIO adapter including flat ribbon cable
- Quick Start card

1.3 Software and documentation

☞ The reference drivers for the ELLA-W1 module series are developed by Marvell and can be re-distributed by u-blox to customers free of charge after signing a license agreement [1]. Please contact u-blox support to obtain the software package.

1.4 System requirements

- Host (PC or embedded system) with SDIO 2.0 capable, full-size SD card socket
- Operating System: Linux (2.6.x/3.x) or Android (4.4)

1.5 Specifications

Table 2 and Table 3 list the absolute maximum ratings and operating conditions for the EVB-ELLA-W1 evaluation board.

Parameter	Description	Min.	Max.	Unit
3V3	Power supply voltage 3.3 V	-0.3	3.6	V
1V8	Power supply voltage 1.8 V	-0.3	2.0	V
VIO	I/O supply voltage 1.8 V/3.3 V	-0.3	3.6	V
$T_{STORAGE}$	Storage temperature	-40	+85	°C

Table 2: Absolute maximum ratings for the EVB-ELLA-W1

Parameter	Description		Min.	Typ	Max.	Unit
3V3	Power supply voltage 3.3 V		3.1	3.3	3.6	V
1V8	Power supply voltage 1.8 V		1.74	1.8	1.89	V
VIO	I/O supply voltage	1.8 V	1.62	1.8	1.98	V
		3.3 V	3.0	3.3	3.6	V
T_A	Ambient operating temperature		-40	-	+85	°C
Ripple Noise	Peak-to-peak voltage ripple on 3V3 and 1V8 supply lines		-	-	10	mV

Table 3: Operating conditions for the EVB-ELLA-W1

2 Getting started

This section describes the evaluation board connectors and configuration settings required to get started.

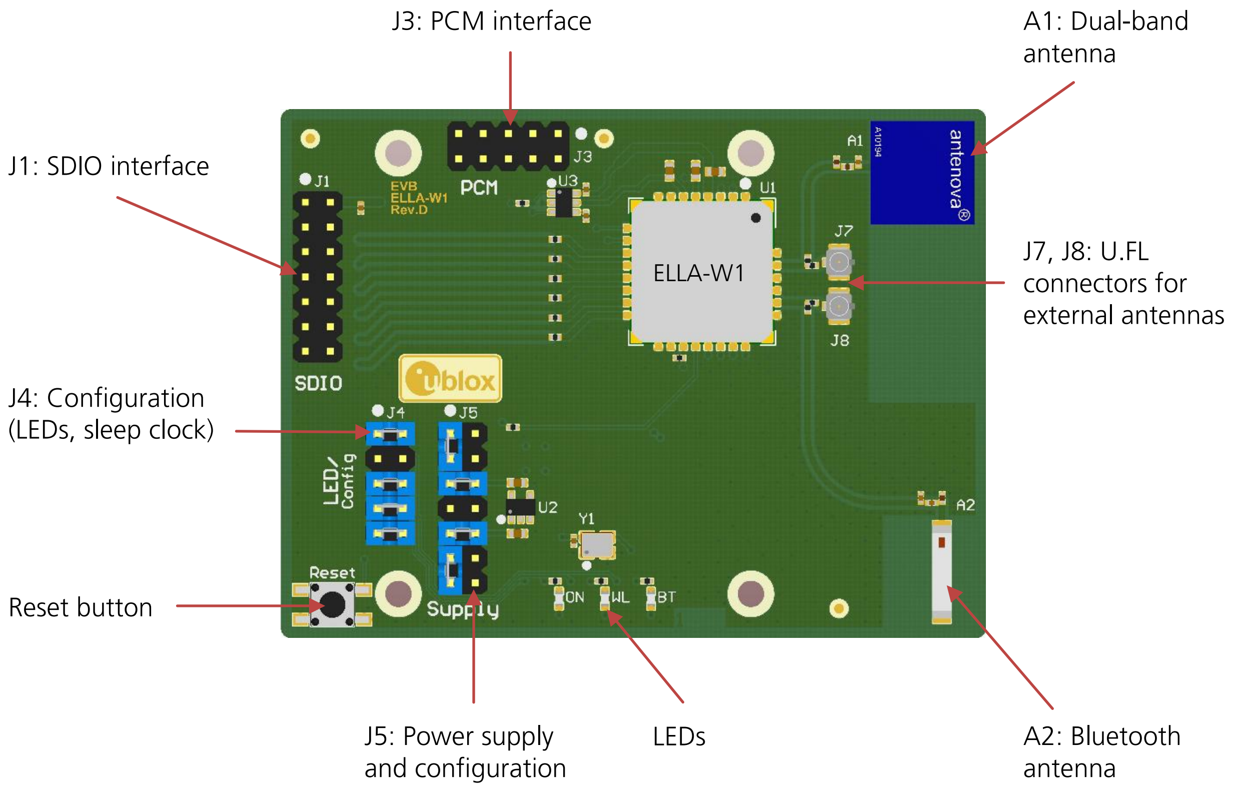

Figure 2: Evaluation board EVB-ELLA-W1 with default jumper settings

Figure 2 shows an overview of the evaluation board and its connectors. The EVK-ELLA-W161 shares the dual-band antenna A1 for Wi-Fi and Bluetooth communication, while the EVK-ELLA-W163 uses the separate antenna A2 for Bluetooth. With the default jumper settings, as shown in Figure 2, power supply from the SDIO interface is used for power supply of the module and the board and I/O voltage is set to 3.3 V. Connect the included SDIO adapter to connector J1 on the evaluation board via the ribbon cable as shown in Figure 1.

2.1 Connecting the evaluation board to the host

Connect the evaluation board to an SDIO capable host by inserting the adapter into the SD card slot. As the ELLA-W1 series module uses an SDIO host interface, only an SDIO capable card reader (not just a common SD card reader) will be able to transfer the data and interrupts correctly. You can use either a built in reader (usually found in laptops – but not all models support SDIO), or a separate reader in one of the extension slots.

An example card reader for Linux is the "Sonnet SDXC UHS-I Pro Reader/Writer ExpressCard/34" [5].

Be careful while inserting the SDIO adapter of the EVK-ELLA-W1 into the SDIO slot of a laptop. Such built-in readers might be designed poorly and can be damaged easily compared to the ones found on development platforms, which are more compact.

The next step is to install the necessary driver software for the ELLA-W1 series modules as described in section 4 of this document.

3 Board description

This section describes the EVB-ELLA-W1 evaluation board and the available connectors and configuration settings.

3.1 Block diagram

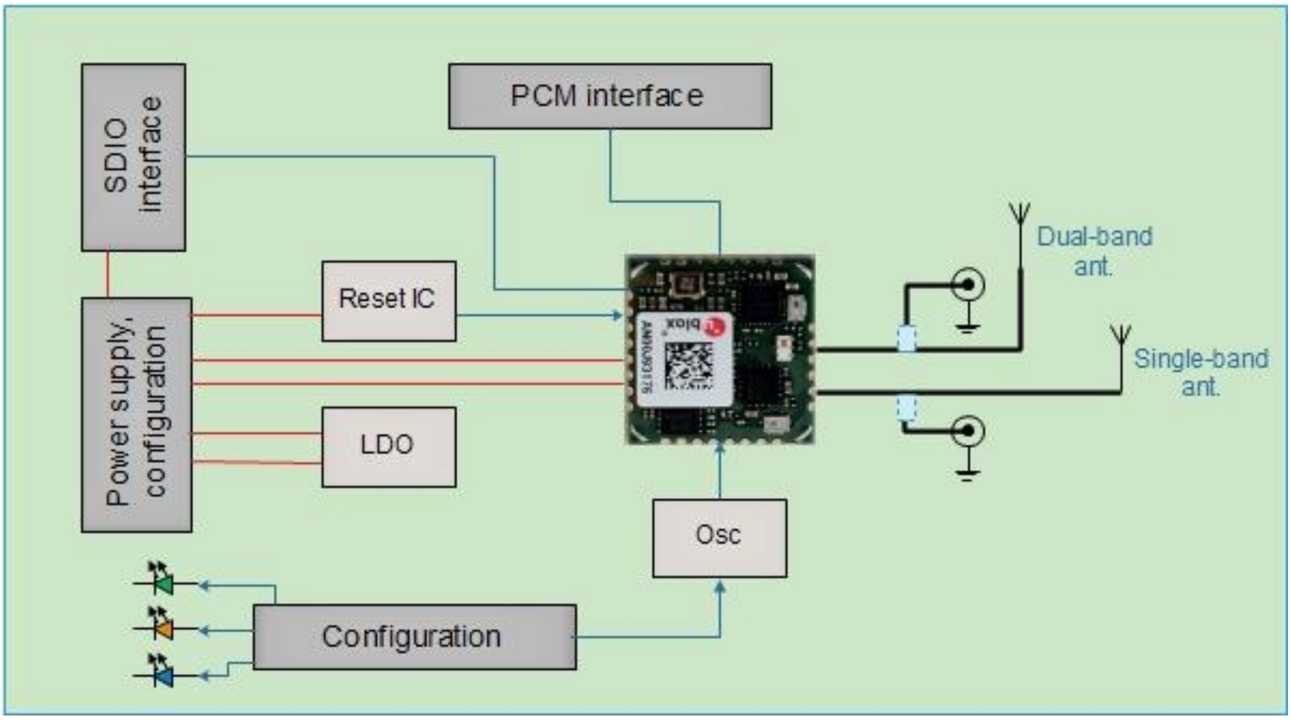

Figure 3: Block diagram of the EVB-ELLA-W1 evaluation board

3.2 Power supply

Different power supply options can be applied to the board using the dual-row pin header J5, which allows to measure current and do performance tests under varying supply conditions. All the required voltages (3V3, 1V8 and VIO) can be supplied from the SDIO bus or from external power supplies. An on-board LDO can generate 1.8 V from a 3.3 V input. VIO voltage can be selected between 1.8 V and 3.3 V.

Table 4 lists the available power supply configuration options.

⚠ **Be careful when configuring the power supply settings, as wrong configurations can cause short circuits and damage the evaluation board and the host system.**

Configuration setting (J5)	Power supply option
11-13 bridged	3V3 supplied from SDIO bus (default)
11-13 open external 3.3 V supply on 11-12	3V3 from external 3.3 V supply
9-10 bridged 5-6 bridged	1V8 supplied from on-board LDO (default)
9-10 open external 1.8 V supply on 5-6	1V8 from external 1.8 V supply
1-3 bridged	VIO voltage set to 3.3 V (default)
1-2 bridged	VIO voltage set to 1.8 V

Table 4: Power supply configuration options

3.3 Configuration

The dual-row pin header J4 is used for configuration purposes and to access digital I/O pins of the ELLA-W1 module as detailed in Table 5.

Configuration setting (J4)	Description
1-2 bridged	ELLA-W1 digital I/O pin LED_1 is used to drive the blue LED
1-2 open	Access to ELLA-W1 digital I/O pin LED_1 on pin 2
3-4 bridged	Pull down ELLA-W1 digital I/O pin CFG; Use of sleep clock is disabled
3-4 open	ELLA-W1 digital I/O pin CFG is left open; Operation with sleep clock is configured
5-6 bridged	ELLA-W1 digital I/O pin LED_0 is used to drive the orange LED
5-6 open	Access to ELLA-W1 digital I/O pin LED_0 on pin 5
7-8 bridged	Green LED connected to 3V3, indicating active supply
7-8 open	Green LED disconnected; LED will not contribute to overall current consumption
9-10 bridged	On-board oscillator for 32.768 kHz sleep clock is powered from 1V8 supply
9-10 open	On-board oscillator for 32.768 kHz sleep clock is not powered for sleep clock-less operation (requires 3-4 to be bridged)

Table 5: Configuration pin header

3.4 Connectors

Table 6 lists the available connectors on the EVB-ELLA-W1 evaluation board and their functions. Refer to the schematic in section 3.7 for details on the pin assignment.

Function	Description	Name
SDIO host interface	Pin header for connecting the SDIO adapter board	J1/SDIO
PCM interface	Pin header for digital audio PCM interface for voice applications	J3/PCM
Primary external antenna connector	U.FL coaxial connector for external 2.4/5 GHz Wi-Fi antenna (and Bluetooth on EVB-ELLA-W161); disconnected by default – populate R2 instead of R1 to use it	J7
Secondary external antenna connector	U.FL coaxial connector for external 2.4 GHz Bluetooth antenna (only used on EVB-ELLA-W163); disconnected by default – populate R10 instead of R9 to use it	J8

Table 6: EVB-ELLA-W1 Connectors description

3.5 LEDs

Table 7 lists the available LEDs on the EVB-ELLA-W1 evaluation board. In the default configuration, the LEDs are used for status indication of the power supply and for showing Wi-Fi/Bluetooth activity.

Function	Description	Name	Color
Main power	Main power supply (3V3)	D1	Green
Wi-Fi activity	Blinking LED shows Wi-Fi Rx/Tx activity (depends on firmware)	D2	Orange
Bluetooth activity	Blinking LED shows Bluetooth Rx/Tx activity (depends on firmware)	D3	Blue

Table 7: LED description

3.6 Reset button

The reset button (S4) on the evaluation board resets the ELLA-W1 module.

3.7 Schematic

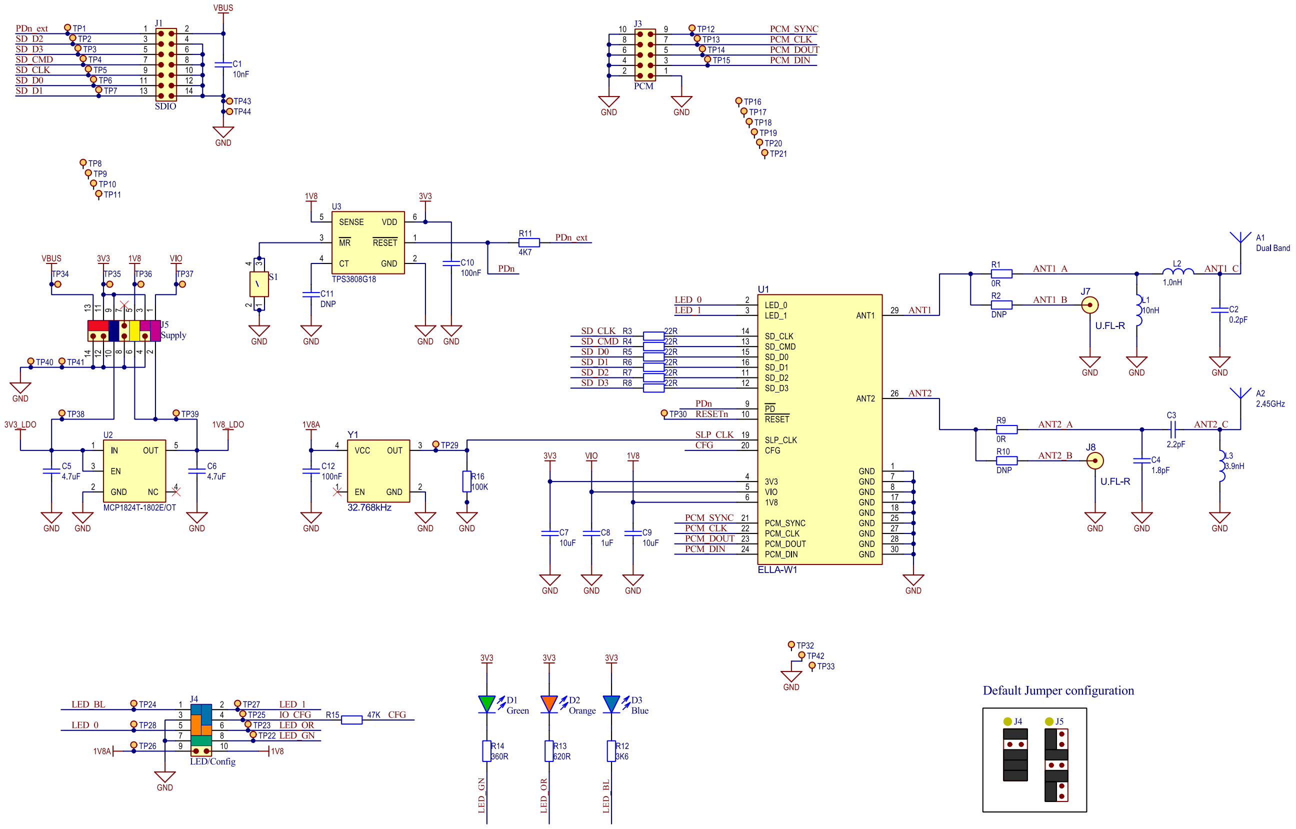

Figure 4: Schematic of the EVB-ELLA-W1 evaluation board

3.8 Assembly

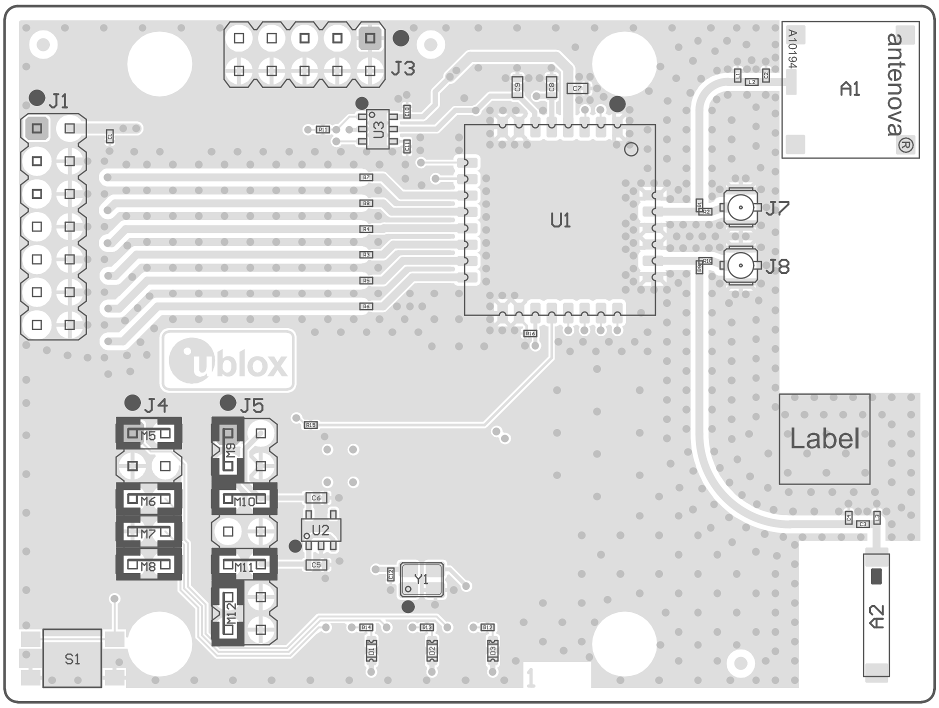

Figure 5: Assembly of the EVB-ELLA-W1 evaluation board

4 Software

The ELLA-W1 module series is based on the Marvell Avastar 88W8787 chipset and it supports both simultaneous and independent operations of:

- Wi-Fi 802.11a/b/g/n (simultaneous client/station, access point and Wi-Fi Direct operation)
- Bluetooth v3.0+HS (also compliant with Bluetooth v2.1+EDR).

The ELLA-W1 modules connect to the host processor through an SDIO interface.

From the software point of view, the ELLA-W1 series modules contain only calibration data and basic operation settings in an on-board EEPROM and thus require a host-side driver and a firmware to run. Each base software package contains the following:

- A firmware image that has to be downloaded to the module on system start and
- A driver, which is placed between the bus driver(s) and the attached network stacks.

Various control tools are also included optionally.

4.1 Linux driver branches

Currently, the following three different driver branches are available for the Linux operating system:

- Marvell automotive drivers (firmware versions 14.44.xx)
- Marvell non-automotive Linux/Android drivers (firmware versions 14.66.xx)
- Open source drivers (mwifiex driver from Linux-mainline)

⚠ **The Software section of this manual describes only the Marvell reference drivers, which can be obtained through u-blox support. The open source drivers are not officially supported by u-blox.**

The automotive and non-automotive driver branches support the main features like parallel access point, station and Wi-Fi Direct operation, and Bluetooth. Refer to the Release Notes that is bundled with each driver release for a list of supported driver features. Generally it is recommended to use the robust automotive drivers for non-automotive projects.

4.1.1 Sleep clock restrictions

The following conditions are applicable for the **automotive firmware**:

- No 32 kHz sleep clock is required.
- If no sleep clock is designed in, then the automotive driver/firmware release must be used, as this alone supports a configuration without sleep clock.

The following conditions are applicable for **general purpose applications**:

- Use of an external 32 kHz sleep clock is mandatory.
- Any driver/firmware release can be used with the above configuration.

The 32 kHz sleep clock is used for Wi-Fi and Bluetooth low-power modes. Hence, a design without an external sleep clock will have restricted Wi-Fi power saving capabilities and no Bluetooth power saving modes. In this case, the automotive firmware release must be used.

4.2 Driver and firmware architecture

The software for the ELLA-W1 modules is split into the following parts:

- The Wi-Fi and Bluetooth driver, running on the host system
- The device firmware, which runs on the module itself

The host drivers interface with the bus drivers and upper layer protocol stacks of the Linux system.

4.2.1 Wi-Fi driver

The basic architecture of the Wi-Fi subsystem is typical of a thick firmware architecture, where the Wi-Fi firmware handles all 802.11 MAC management tasks.

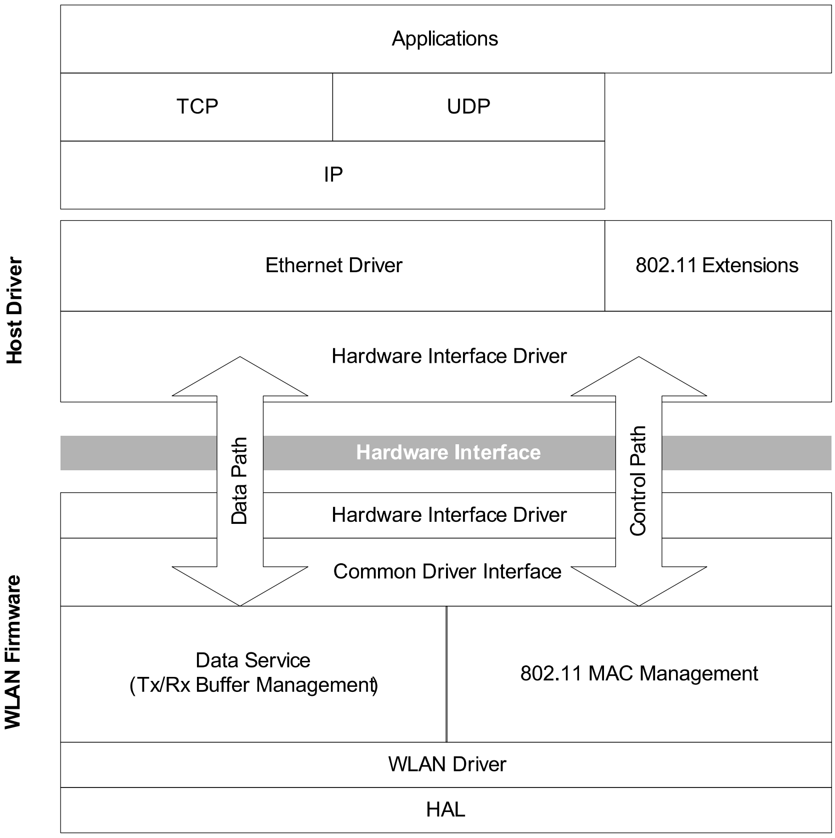

Figure 6: Basic Wi-Fi driver and firmware architecture

The following steps are performed while loading the Wi-Fi host driver:

- The driver registers itself with the MMC/SDIO bus driver.
- Upon successful registration, the bus driver calls the Wi-Fi driver's probe handler, when the module is detected.
- The probe handler allocates and initializes internal structures, registers the interrupt service routine and starts the main driver threads.
- The firmware image is downloaded to the module and the hardware is initialized.
- Network devices such as STA, AP, and WFD are registered.

4.2.2 Bluetooth driver

The standard Bluetooth protocol stack in Linux is provided by BlueZ. The Bluetooth driver for the ELLA-W1 module series is a client driver that runs on top of the MMC/SDIO bus driver and it does the following:

- Forwards the data and commands between upper protocol stack layers and the firmware.
- Handles some private commands that are used as handshake between the driver and firmware only.

On loading the driver, it registers with the bus driver, downloads the firmware, if not already loaded by the Wi-Fi driver, and registers a new HCI device with the BlueZ stack.

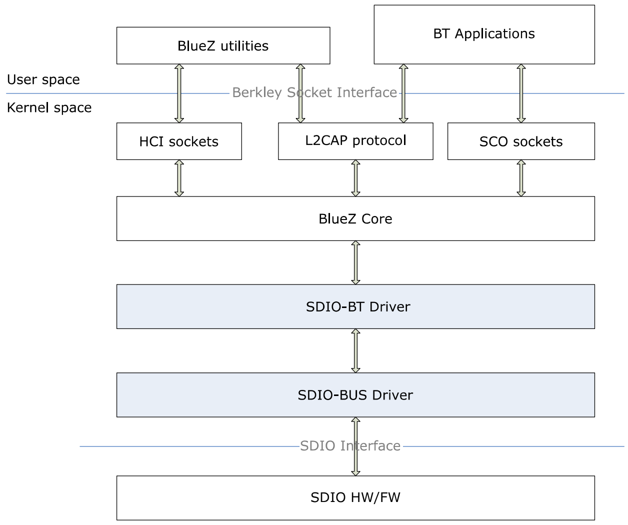

Figure 7: Bluetooth protocol stack

Another driver for Bluetooth is also available in the reference driver package, which is not bound to BlueZ, instead it exports a character device, which can be used by third-party user space Bluetooth stacks.

4.3 Compiling the drivers

4.3.1 Prerequisites

4.3.1.1 Reference drivers

The versions of the Marvell Linux reference drivers/firmware package and the Linux OS that are used for this document are:

- Marvell Linux reference driver 14.44.35.p233-M2614525 (automotive version)
- Linux 3.19.8

The drivers should be able to support Linux kernel versions from 2.6.31 to 4.1. Older or more recent kernels might require some patches due to changed kernel APIs. Patches for compiling the Marvell driver branches on the u-blox EVK-W16 reference platform, which is currently running a 3.19.8 kernel, can be provided on request.

☞ The reference drivers for the ELLA-W1 module series are developed by Marvell and can be re-distributed by u-blox to customers after signing a license agreement [1].

4.3.1.2 Kernel configuration

The drivers for the ELLA-W1 series modules depend on the MMC/SDIO stack of the Linux kernel; thus it must be enabled on the target system. For configuration, the Linux reference driver supports the following two driver API options:

- The old Linux wireless extensions (WEXT) interface
- The new cfg80211 configuration API

To enable these APIs on the target system, the following must be selected in the kernel configuration (`CONFIG_WIRELESS_EXT` cannot be selected directly, so a driver that depends on it, such as hostap or zd1201 must be selected):

```
CONFIG_WIRELESS_EXT=y
CONFIG_WEXT_PRIV=y
CONFIG_CFG80211=y
```

Listing 1: Kernel .config

⚠ **For older kernels (<3.2), use compat-wireless (now named backports) to provide recent versions of the kernel's 802.11 APIs to support all the driver features. In this case, cfg80211 has to be compiled as a module (CONFIG_CFG80211=m).**

4.3.2 Extracting package content

The Marvell driver package contains the firmware image, the Wi-Fi/Bluetooth driver sources and also a release notes that describes the tested hardware platform, supported features, bug fixes and known limitations of the release. The package comes as several archives that are packed into each other. Follow the steps mentioned below to extract the Marvell driver package:

```
unzip SD-UAPSTA-BT-FM-8787-FC13-MMC-14.44.35.p233-M2614525_AX-GPL.zip
tar xf SD-UAPSTA-BT-FM-8787-FC13-MMC-14.44.35.p233-M2614525_AX-GPL.tar
for i in *.tgz; do tar xzf $i; done
```

Once you remove the archives, you should find something similar to the following in your working directory:

```
├── FwImage/
│   └── sd8787_uapsta.bin         # binary firmware image
└── SD-UAPSTA-BT-FM-8787-FC13-MMC-14.44.35.p233-M2614525_AX-GPL/
    ├── mbtc_src/                 # character device driver for 3rd party user space Bluetooth stacks
    ├── mbt_src/                  # driver for the Linux Bluetooth stack bluez
    └── wlan_src/                 # Wi-Fi driver and tools sources
        ├── Makefile
        ├── mapp/                 # user space tools for configuration, sample config files
        ├── mlan/                 # OS independent driver sources
        ├── mlinux/               # Linux specific driver sources
        └── [...]
```

4.3.3 Compile-time configuration

The Wi-Fi driver has several compile-time configuration options that can be set in the driver's Makefile. Change to the `wlan_src` subdirectory and ensure that the following are enabled:

```
# Enable STA mode support
CONFIG_STA_SUPPORT=y
# Enable uAP mode support
CONFIG_UAP_SUPPORT=y
# Manufacturing firmware support
CONFIG_MFG_CMD_SUPPORT=y
```

Listing 2: Makefile

The manufacturing firmware support is required, if the driver is used with the "Manufacturing and Labtools" packages, which can be used for setting up test modes for certification [4].

4.3.4 Building

4.3.4.1 Prepare kernel sources

Primarily, ensure that your kernel is prepared for compiling external kernel modules. For this, change to the kernel's source directory and run the following:

```
make modules_prepare
```

⚠ **"make modules_prepare" will not build Module.symvers even if CONFIG_MODVERSIONS is set; therefore, a full kernel build must be executed to make module versioning work.**

4.3.4.2 Wi-Fi driver and tools

To compile the Wi-Fi drivers and tools, go to the `wlan_src` subdirectory in driver packages and run '`make build`'. For cross-compilation, you should specify the target architecture, cross-toolchain prefix and the directory with the kernel sources used to build the kernel on the target system, that is:

```
# e.g.:
#  ARCH=arm
#  CROSS_COMPILE=arm-poky-linux-gnueabi-
#  KERNELDIR=/home/user/work/linux-3.19.8/

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

This command will build the Wi-Fi kernel modules and all the included user space applications. The build results will be copied to `../bin_sd8787/`, relative to the `wlan_src` directory. The following table summarizes the content of the Wi-Fi build results directory:

File	Description
mlan.ko, sd8787.ko	Wi-Fi driver kernel modules
README*	Usage instructions for the provided tools
config/*	Sample configuration files used by various tools
mlanevent.exe	Netlink event listener
wifidirectutl	Configures Wi-Fi Direct parameters
uaputl.exe	Configures micro-AP settings
mlanutl	Configures additional driver parameters
mlan2040coex	802.11 20/40 MHz coexistence handler

Table 8: Content of the Wi-Fi build results directory

4.3.4.3 Bluetooth driver and tools

To compile the BlueZ Bluetooth driver and tools, go to the driver packages's `mbt_src` subdirectory and run `'make build'` again (see Wi-Fi driver for cross-compilation). The build results will be copied to the directory..`/bin_sd8787_bt/.` The following table summarizes the content of the Bluetooth build results directory:

File	Description
bt8787.ko	Bluetooth driver kernel module
fmapp	
config/*	Sample configuration files
README	Usage instructions

Table 9: Content of the Bluetooth build results directory

4.4 Deploying the software

The following steps describe how to install the drivers, firmware, and provided tools on the target system:

1. Copy the application binaries to an appropriate location on the target file system and add it to the `$PATH` environment variable, if required.
2. The kernel modules should be copied to somewhere below the modules directory of the kernel, for example, `/lib/modules/3.19.8/updates/`. Run the depmod command afterwards to update the module dependencies and to have the modules findable by the modprobe utility.

3. Copy the firmware image file `sd8787_uapsta.bin` from the driver package's `FwImage` directory to the directory `/lib/firmware/mrvl/` on the target file system.

An example deployment is shown below:

```
/
├── lib
│   ├── firmware
│   │   └── mrvl
│   │       └── sd8787_uapsta.bin
│   └── modules
│       └── 3.19.8
│           └── updates
│               ├── mlan.ko
│               ├── sd8787.ko
│               └── bt8787.ko
└── opt
    └── ella-w1
        ├── bin_sd8787
        └── bin_sd8787_bt
```

Listing 3: Example target file system

4.4.1 Blacklisting the mwifiex driver

If the target system includes the open source mwifiex driver, make sure to use the correct firmware image by replacing the existing one and that the mwifiex driver is blacklisted to prevent it from being loaded automatically. To blacklist the mwifiex kernel modules, add the following lines to a file under `/etc/modprobe.d/`, for example in `/etc/modprobe.d/blacklist.conf`:

```
blacklist mwifiex
blacklist mwifiex_sdio
```

Listing 4: Blacklisting mwifiex

⚠ **Blacklisting will not work for drivers that are built into the kernel image rather than as a kernel module.**

4.4.2 Additional software requirements

Some additional packages that are recommended for installation on the target system are mentioned in the following table:

Package	Comment
bluez4 or bluez5	Contains the user space parts of the Linux Bluetooth stack
wpa_supplicant	WPA supplicant. Handles key negotiation and roaming etc. on client side
iw	CLI configuration utility for wireless devices
wireless-tools	CLI tools for configuring wireless device drivers using Wireless Extensions
crda	User space udev helper to handle regulatory domain

Table 10: Recommended additional software packages

4.5 Loading the drivers

4.5.1 Wi-Fi

If the kernel modules were installed correctly, you can load them by simply issuing the following command

```
modprobe sd8787 cfg80211_wext=0xf
```

Else, you have to load them separately using the insmod command.

This will automatically load the sd8787 kernel module and all its dependencies, such as mlan or cfg80211. The `cfg80211_wext=0xf` module parameter in the above-mentioned example informs the driver to enable support for the wireless extensions interface and for the cfg80211 configuration API. A full description of the available module parameters is given in the README files and also in the '`modinfo sd8787`' command. If the drivers are successfully loaded, you should see them in the list of loaded modules as shown below:

☞ The internal name for the sd8787 module is sd8xxx.

```
Module      Size  Used by
sd8xxx    383837  0
mlan      284052  1 sd8xxx
```

Listing 5: lsmod output

When the module is detected on the SDIO interface, the driver will automatically download the firmware to it, initialize the hardware, and register the network interfaces.

```
mmc1: new high speed SDIO card at address 0001
wlan: Loading MWLAN driver
vendor=0x02DF device=0x9119 class=0 function=1
SDIO: max_segs=1024 max_seg_size=33553920
rx_work=0 cpu_num=1
wlan: Enable TX SG mode
wlan: Enable RX SG mode
Wlan: FW download over, firmwarelen=453696 downloaded 453696
WLAN FW is active
fw_cap_info=0xf03, dev_cap_mask=0xffffffff
wlan: version = SD8787-14.44.35.p233-M2614525-GPL-(FP44)
wlan: Driver loaded successfully
```

Listing 6: Kernel log after inserting the SDIO card

You should be able to see the following new network interfaces (for example using the '`ifconfig -a`' or '`iw dev`' commands):

Interface	Function
mlan0	Wi-Fi station mode
uap0	Wi-Fi micro access point mode
wfd0	Wi-Fi Direct

The version of the loaded firmware can be verified for example, by using one of the following commands:

```
$ mlanutl mlan0 version
Version string received: SD8787-14.44.35.p233-M2614525-GPL-(FP44)
$ iwpriv mlan0 version
mlan0     version:SD8787-14.44.35.p233-M2614525-GPL-(FP44)
```

4.5.2 Bluetooth

To load the Bluetooth driver:

```
modprobe bt8787
```

This will register a new Bluetooth device (hci0 in this case). The firmware download will be skipped if the Wi-Fi driver has already been loaded.

```
$ hciconfig
hci0:     Type: BR/EDR  Bus: SDIO
          BD Address: 00:06:C6:46:DF:7B  ACL MTU: 1021:6  SCO MTU: 120:6
          UP RUNNING PSCAN
          RX bytes:656 acl:0 sco:0 events:28 errors:0
          TX bytes:986 acl:0 sco:0 commands:28 errors:0
```

4.5.3 Unloading the drivers

To unload the drivers, bring all the interfaces down first and then remove the modules using:

```
rmmod mlan sd8xxx bt8xxx
```

4.6 Usage examples

4.6.1 Wi-Fi access point mode

The following example configures and starts an access point using the provided Marvell tools. A more detailed description of the uaputl.exe tool and its parameters can be found in the README_UAP file from the driver package.

```
uaputl.exe sys_cfg_ssid ELLA-W1                                   # set AP SSID to "ELLA-W1"
# set AP primary channel to 36 (5GHz band), with secondary channel above:
uaputl.exe sys_cfg_channel 36 2
# enable 802.11n mode with short guard interval, 40MHz channel bandwidth:
uaputl.exe sys_cfg_11n 1 0x116e 3 0 0xff
uaputl.exe sys_cfg_rates 0xc 0x12 0x18 0x24 0x30 0x48 0x60 0x6c
uaputl.exe sys_cfg_80211d state 1 country US                      # enable 802.11d, set country

# configure encryption:
uaputl.exe sys_cfg_auth 0
uaputl.exe sys_cfg_protocol 32                                    # WPA2
uaputl.exe sys_cfg_wpa_passphrase topsecret                       # passphrase "topsecret"
uaputl.exe sys_cfg_cipher 8 8                                     # CCMP

uaputl.exe bss_start                                              # start the AP
```

Listing 7: Create a Wi-Fi access point

To assign an IP address to the access point interface:

```
ifconfig uap0 192.168.1.1
```

Additionally, it is recommended to use a DHCP server on the interface.

4.6.2 Wi-Fi station mode

4.6.2.1 Using Marvell tools

This example will connect the ELLA module as a station to an access point. A description of the used commands and parameters can be found in the provided README and README_MLAN files.

```
mlanutl mlan0 countrycode US                                        # set countrycode
mlanutl mlan0 passphrase "1;ssid=MyAP;passphrase=12345678"          # set passphrase for WPA/WPA2
mlanutl mlan0 reassoctrl 1                                          # turn on re-association
mlanutl mlan0 assocessid MyAP                                       # connect to AP with SSID "MyAP"
udhcpc -i mlan0                                                     # request IP address per DHCP
```

Listing 8: Connect to an access point (AP) in station mode

4.6.2.2 Using wpa_supplicant

It is also possible to let wpa_supplicant handle the connection to the access point. For this, create a configuration file containing the following network settings:

```
ctrl_interface=/var/run/wpa_supplicant
ap_scan=1
network={
    scan_ssid=1
    ssid="MyAP"
    key_mgmt=WPA-PSK
    psk="12345678"
}
```

Listing 9: wpa_supplicant.conf

Then run the wpa_supplicant daemon using the configuration file:

```
wpa_supplicant -D nl80211 -i mlan0 -c /etc/wpa_supplicant.conf –B
```

To configure the IP address through DHCP:

```
udhcpc -i mlan0
```

4.6.3 Wi-Fi Direct

Refer to ELLA-W1 Wi-Fi Direct application note [3].

4.6.4 Bluetooth

After loading the Bluetooth driver, an HCI device should be available. To use the HCI device, the interface must be set up first. This can be done using the following hciconfig command from the BlueZ package:

```
hciconfig hci0 up
```

To enable any Bluetooth services, the Bluetooth daemon should be started:

```
bluetoothd
```

To verify if the Bluetooth is working, you can issue a scan request to search for remote devices and try to ping them using L2CAP echo requests:

```
$ hcitool -i hci0 scan
Scanning ...
        00:22:58:F8:86:BB       ae-sho-bln-test
$ l2ping -i hci0 00:22:58:F8:86:BB
Ping: 00:22:58:F8:86:BB from 00:06:C6:46:DF:7B (data size 44) ...
4 bytes from 00:22:58:F8:86:BB id 0 time 69.75ms
4 bytes from 00:22:58:F8:86:BB id 1 time 56.76ms
[...]
```

4.7 Driver debugging

Driver debugging is provided via the kernel print function `printk` and the `proc` file system. The driver states are recorded and can be retrieved through the proc file system during runtime. The following files containing the debug information are provided (the actual location is dependent on the Linux kernel version):

- /proc/mwlan/config or /proc/net/mwlan/config
- /proc/mwlan/mlanX/info or /proc/net/mwlan/mlanX/info
- /proc/mwlan/mlanX/debug or /proc/net/mwlan/mlanX/debug

☞ mlanX is the name of the device node created at runtime. Other possibilities are uapX and wfdX for the acces point and Wi-Fi Direct interfaces respectively.

Debug messages are also printed to the kernel ring buffer through printk calls. These messages can be accessed raw using the `/proc/kmsg` interface or by the dmesg command. Alternatively, this can also be handled by more advanced logging facilities.

4.7.1 Compile-time debug options

The extent to which debug messages are available for printing at runtime is controlled by the `CONFIG_DEBUG` variable in the driver's Makefile. The `CONFIG_DEBUG` variable can have the following values:

- n: debug messages are disabled and not compiled into the driver module
- 1: all kinds of debug messages can be configured except for MENTRY, MWARN and MINFO. By default MMSG, MFATAL and MERROR are enabled.
- 2: all kinds of debug messages can be configured

4.7.2 Runtime debug options

Once debugging is enabled in the Makefile, debug messages can be selectively enabled or disabled at runtime by setting or clearing the corresponding bits of the `drvdbg` parameter:

```
bit 0:  MMSG          PRINTM(MMSG,...)
bit 1:  MFATAL        PRINTM(MFATAL,...)
bit 2:  MERROR        PRINTM(MERROR,...)
bit 3:  MDATA         PRINTM(MDATA,...)
bit 4:  MCMND         PRINTM(MCMND,...)
bit 5:  MEVENT        PRINTM(MEVENT,...)
bit 6:  MINTR         PRINTM(MINTR,...)
bit 7:  MIOCTL        PRINTM(MIOCTL,...)
...
bit 16: MDAT_D        PRINTM(MDAT_D,...), DBG_HEXDUMP(MDAT_D,...)
bit 17: MCMD_D        PRINTM(MCMD_D,...), DBG_HEXDUMP(MCMD_D,...)
bit 18: MEVT_D        PRINTM(MEVT_D,...), DBG_HEXDUMP(MEVT_D,...)
bit 19: MFW_D         PRINTM(MFW_D,...),  DBG_HEXDUMP(MFW_D,...)
bit 20: MIF_D         PRINTM(MIF_D,...),  DBG_HEXDUMP(MIF_D,...)
...
bit 28: MENTRY        PRINTM(MENTRY,...), ENTER(), LEAVE()
bit 29: MWARN         PRINTM(MWARN,...)
bit 30: MINFO         PRINTM(MINFO,...)
```

The value of **drvdbg** can be given as a module parameter when the driver is loaded, by writing to the proc file system's **debug** file or by setting it via the iwpriv or mlanutl tool.

```
iwpriv mlan0 drvdbg                                               # Get the current driver debug mask
iwpriv mlan0 drvdbg 0                                             # Disable all debug messages
echo "drvdbg=0x7" > /proc/mwlan/mlan0/debug                       # enable MMSG, MFATAL and MERROR
mlanutl mlan0 drvdbg -1                                           # Enable all debug messages
```

Listing 10: Debug examples

Appendix

A Glossary

Name	Definition
AP	Access point
API	Application Programming Interface
DHCP	Dynamic Host Configuration Protocol
EDR	Enhanced Data Rate
EEPROM	Electrically Erasable Programmable Read-Only Memory
HCI	Host Controller Interface
LED	Light-Emitting Diode
MAC	Medium Access Control
MMC	Multimedia Card
OS	Operating System
SDIO	Secure Digital Input Output
STA	Station
SSID	Service Set Identifier
uAP	Micro Access Point
WEXT	Wireless Extensions
WFD	Wi-Fi Direct
Wi-Fi	Wireless Local Area Network
WPA	Wi-Fi Protected Access

Related documents

[1] u-blox Limited Use License Agreement for Marvell SW platform and OSS Deliverables

[2] ELLA-W1 series Data sheet, Docu.No. UBX-15004476

[3] ELLA-W1 Wi-Fi Direct Application note, Docu.No. UBX-15014432

[4] Radio Test Guide Application Note, Docu.No. UBX-15014433

[5] Sample card reader for Linux: Sonnet SDXC UHS-I Pro Reader/Writer ExpressCard/34 - http://www.sonnettech.com/product/sdxcproreader.html

Revision history

Revision	Date	Name	Status / Comments
R01	27-May-2015	mzes	Initial release.
R02	1-July-2015	lalb, mzes	Minor updates.
R03	4-May-2016	mzes, lalb, kgom	Updated to new evaluation board hardware revision (EVK type numbers ending with "-01"). Included the different evaluation kit variants of EVK-ELLA-W1 in table format (Table 1). Included the preliminary version of the EVK-ELLA-W1 board in Figure 1 and included a footnote to this effect. Updated the version of the automotive driver package and changed instructions to use this as an example.
R04	5-Oct-2016	mzes	Changed the document status to Early Production Information.

Contact

For complete contact information visit us at www.u-blox.com.

u-blox Offices

North, Central and South America

u-blox America, Inc.

Phone:	+1 703 483 3180
E-mail:	info_us@u-blox.com

Regional Office West Coast:

Phone:	+1 408 573 3640
E-mail:	info_us@u-blox.com

Technical Support:

Phone:	+1 703 483 3185
E-mail:	support_us@u-blox.com

Headquarters
Europe, Middle East, Africa

u-blox AG

Phone:	+41 44 722 74 44
E-mail:	info@u-blox.com
Support:	support@u-blox.com

Asia, Australia, Pacific

u-blox Singapore Pte. Ltd.

Phone:	+65 6734 3811
E-mail:	info_ap@u-blox.com
Support:	support_ap@u-blox.com

Regional Office Australia:

Phone:	+61 2 8448 2016
E-mail:	info_anz@u-blox.com
Support:	support_ap@u-blox.com

Regional Office China (Beijing):

Phone:	+86 10 68 133 545
E-mail:	info_cn@u-blox.com
Support:	support_cn@u-blox.com

Regional Office China (Chongqing):

Phone:	+86 23 6815 1588
E-mail:	info_cn@u-blox.com
Support:	support_cn@u-blox.com

Regional Office China (Shanghai):

Phone:	+86 21 6090 4832
E-mail:	info_cn@u-blox.com
Support:	support_cn@u-blox.com

Regional Office China (Shenzhen):

Phone:	+86 755 8627 1083
E-mail:	info_cn@u-blox.com
Support:	support_cn@u-blox.com

Regional Office India:

Phone:	+91 80 4050 9200
E-mail:	info_in@u-blox.com
Support:	support_in@u-blox.com

Regional Office Japan (Osaka):

Phone:	+81 6 6941 3660
E-mail:	info_jp@u-blox.com
Support:	support_jp@u-blox.com

Regional Office Japan (Tokyo):

Phone:	+81 3 5775 3850
E-mail:	info_jp@u-blox.com
Support:	support_jp@u-blox.com

Regional Office Korea:

Phone:	+82 2 542 0861
E-mail:	info_kr@u-blox.com
Support:	support_kr@u-blox.com

Regional Office Taiwan:

Phone:	+886 2 2657 1090
E-mail:	info_tw@u-blox.com
Support:	support_tw@u-blox.com

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for Interface Development Tools ***category:***

Click to view products by U-Blox ***manufacturer:***

Other Similar products are found below :

CY4607M PEX 8748-CA RDK DP130DSEVM DP130SSEVM ISO3086TEVM-436 SP338EER1-0A-EB ADM00276 ADM3054WBRWZ-RL7
ADP5585CP-EVALZ PEX8724-CA RDK PEX 8732-CA RDK PEX8747-CA RDK PS081-EVA-KIT CHA2066-99F AS8650-DB MLX80104
TESTINTERFACE I2C-CPEV/NOPB ISO35TEVM-434 KIT33978EKEVB 416100120-3 XR17D158CV-0A-EVB XR17V358/SP339-E4-EB
XR17V358/SP339-E8-EB XR18910ILEVB XR22804IL56-0A-EB ZSC31150KIT V1.2 SCRUBBER-EVM SI838XISO-KIT 73931-3022
XIO2200AEVM XIB-E XBIB-U-SP TW-DONGLE-USB EVAL-ADM2483EBZ EVAL-ADM2491EEBZ EVB-USB83340 MAX9921EVKIT
MAXREFDES23DB# MAX9291COAXEVKIT# MAX9286COAXEVKIT# MAX3535EEVKIT+ MAX3223EEVKIT+ MAX3100EVKIT
MAX13235EEVKIT MAX14970EVKIT# MAX148X1EVKIT# MAX14826EVKIT# 3298 XR21B1424IV64-0A-EVB XR21B1421IL24-0A-EVB