EVK-LILY-W1

Evaluation kit for the LILY-W1 host-based Wi-Fi modules

User Guide

Abstract

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

www.u-blox.com

UBX-15030290 - R04

Document Information

Title	**EVK-LILY-W1**	
Subtitle	Evaluation kit for the LILY-W1 host-based Wi-Fi modules	
Document type	User Guide	
Document number	UBX-15030290	
Revision and date	R04	3-Jul-2018
Disclosure Restriction		

This document applies to the following products:

Product name	Type number	Firmware version	PCN reference	Product status
EVK-LILY-W131	EVK-LILY-W131-00	SDIO driver: Package: SD-UAPSTA-8801-FC18-MMC-14.85.36.p101-C3X14160_B0-GPL Firmware version: 14.85.36.p101 USB driver: Package: USB-UAPSTA-8801-FC18-X86-14.85.36.p101-C3X14160_B0-GPL Firmware version: 14.85.36.p101	N/A	Early Production Information
EVK-LILY-W132	EVK-LILY-W132-00	SDIO driver: Package: SD-UAPSTA-8801-FC18-MMC-14.85.36.p101-C3X14160_B0-GPL Firmware version: 14.85.36.p101 USB driver: Package: USB-UAPSTA-8801-FC18-X86-14.85.36.p101-C3X14160_B0-GPL Firmware version: 14.85.36.p101	N/A	

u-blox or third parties may hold intellectual property rights in the products, names, logos and designs included in this document. Copying, reproduction, modification or disclosure to third parties of this document or any part thereof is only permitted with the express written permission of u-blox.
The information contained herein is provided “as is” and u-blox assumes no liability for its use. No warranty, either express or implied, is given, including but not limited to, 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 without notice. For the most recent documents, visit www.u-blox.com.
Copyright © u-blox AG.

Contents

Document Information **2**
Contents **3**
1 Evaluation kit description **5**
1.1 Overview 5
1.2 Kit includes 5
1.3 Software and documentation 5
1.4 System requirements 6
1.5 Specifications 6
2 Getting started **7**
2.1 Host interface through SDIO 8
2.2 Host interface through USB 8
3 Board description **9**
3.1 Connectors 9
3.2 Pin header J2 9
3.3 Pin header J3 10
3.4 Antenna interfaces on the EVB-LILY-W131 10
3.4.1 Single antenna 10
3.4.2 Antenna diversity 11
3.4.3 LTE filter 11
3.5 LED 11
3.6 Schematic 12
3.7 Assembly 12
4 Software **13**
4.1 Driver versions 13
4.2 Driver and firmware architecture 13
4.3 Compiling the drivers 14
4.3.1 Prerequisites 14
4.3.2 Extracting the package content 15
4.3.3 Compile-time configuration 15
4.3.4 Building 16
4.4 Deploying the software 16
4.4.1 Blacklisting the mwifiex driver 17
4.4.2 Additional software requirements 17
4.5 Loading the drivers 17
4.5.1 SDIO driver 17
4.5.2 USB driver 18
4.5.3 Unloading the drivers 19
4.6 Usage examples 19
4.6.1 Wi-Fi access point mode 19
4.6.2 Wi-Fi station mode 20
4.7 Driver debugging 21

4.7.1 Compile-time debug options 21
4.7.2 Runtime debug options 21
Appendix 23
A Glossary 23
Related documents 24
Revision history 24
Contact 25

1 Evaluation kit description

1.1 Overview

The EVK-LILY-W1 evaluation kit provides a simple way to evaluate the LILY-W1 host-based Wi-Fi modules. The evaluation board offers a standard full-size SDIO card connector (compatible with host sockets designed for SD memory cards) and a micro-USB receptacle for host communication.

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

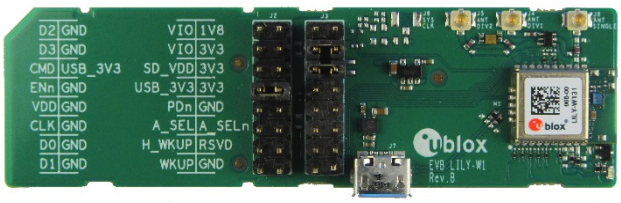
- Available for internal antenna and antenna pin variants of the LILY-W1 module
- SDIO 2.0 device interface via SDIO card connector for host communication
- USB 2.0 device interface via micro-USB receptacle for host communication
- U.FL coaxial connectors for external antenna and antenna diversity (EVK-LILY-W131)
- All module interfaces are externally available
- Multiple power supply options

Table 1 lists the available evaluation kit versions.

Evaluation kit	Description	Suitable for evaluation of
EVK-LILY-W131	Evaluation kit for version with antenna pin; without internal LTE filter	LILY-W131
EVK-LILY-W132	Evaluation kit for version with internal antenna and LTE filter	LILY-W132

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

Figure 1 and Figure 2 show the EVK-LILY-W1 evaluation kits.



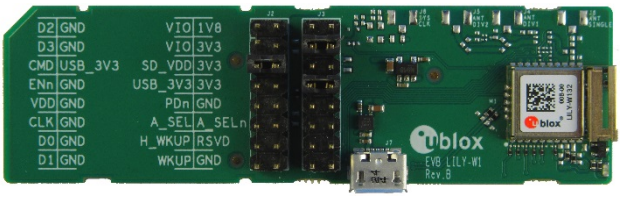


Figure 2: EVK-LILY-W132 evaluation kit

Figure 1: EVK-LILY-W131 evaluation kit including antenna

1.2 Kit includes

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

- Evaluation board (EVB) for LILY-W131 (EVB-LILY-W131) or LILY-W132 (EVB-LILY-W132)
- One 2.4 GHz antenna (ProAnt Ex-IT 2400 RP-SMA 70-002) and RP-SMA to U.FL adapter cable (EVK-LILY-W131 only)
- Quick Start card

1.3 Software and documentation

☞ The reference drivers for the LILY-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]*. 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 or
  - USB 2.0 interface
- 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 LILY-W1 evaluation board.

Parameter	Description	Min.	Max.	Unit
SDCARD.VDD	SD card power supply	-	4.0	V
USB.VBUS	USB power supply	-	6.0	V
$T_{STORAGE}$	Storage temperature	-40	+85	°C

Table 2: Absolute maximum ratings for the LILY-W1 evaluation board

Parameter	Description	Min.	Typ	Max.	Unit
SDCARD.VDD	SD card power supply, 3.3 V operation	3.0	3.3	3.6	V
	SD card power supply, 1.8 V operation	1.65	1.8	1.95	V
USB.VBUS	USB power supply	4.25	5.0	5.5	V
Digital I/O	SDIO interface and other GPIO	-0.4	-	SDCARD.VDD+0.4	V
1.8V Digital I/O	WKUP and ENn pins	-0.4	-	2.2	V
T_A	Ambient operating temperature	-40	-	+85	°C

Table 3: Operating conditions for the LILY-W1 evaluation board

2 Getting started

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

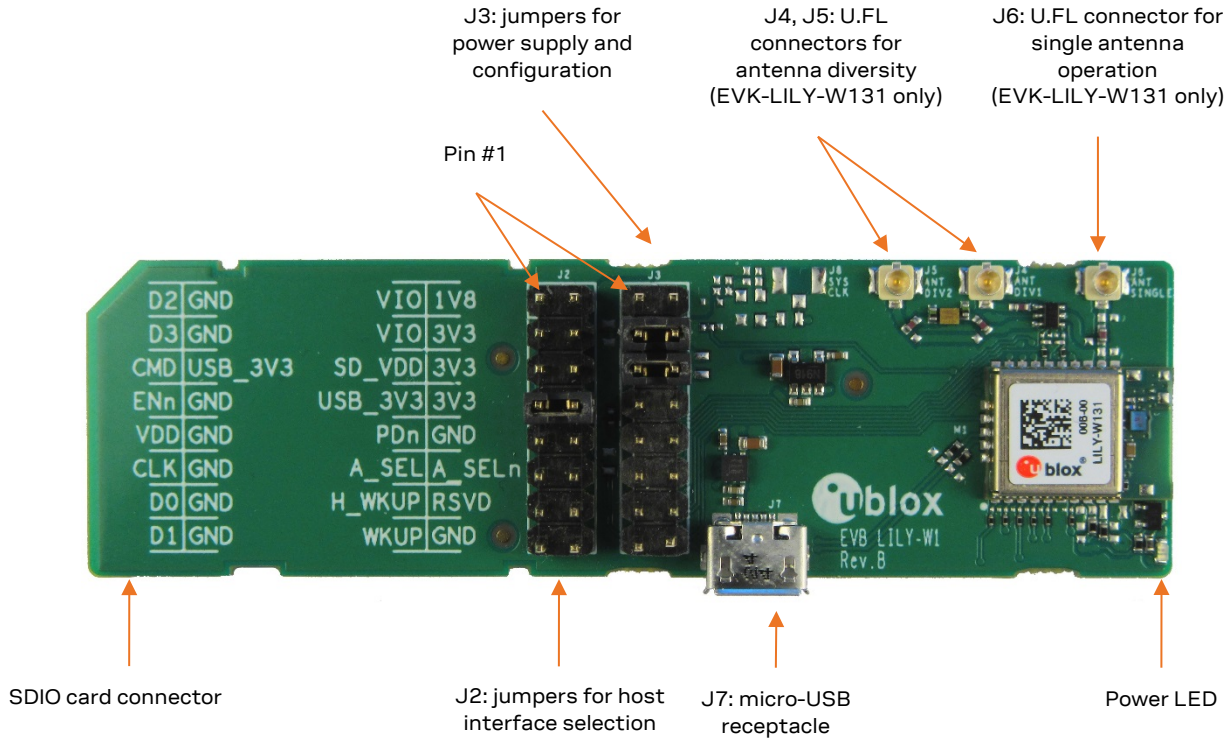


Figure 3: Evaluation board EVB-LILY-W131

Figure 3 shows an overview of the evaluation board and its connectors. The board can be connected to a host system either through the SDIO card connector or the micro-USB connector. The U.FL connectors - J4, J5, and J6 can only be used with the EVK-LILY-W131, which uses the LILY-W131 module version with the antenna pin. The antenna U.FL connector J6 is connected to the module by default. The evaluation of antenna diversity via the U.FL connectors J4 and J5 requires switching of components on the board.

The following steps are required to start the evaluation:

1. For the EVK-LILY-W131, connect the included 2.4 GHz antenna via the adapter cable to the U.FL single antenna connector (J6) on the EVB-LILY-W131 evaluation board. For the EVK-LILY-W132, you need not connect an external antenna as it uses the internal antenna of the LILY-W132 module.
2. Select the SDIO (Figure 4) or USB (Figure 5) interface for host communication as described in sections 2.1 and 2.2 and connect the evaluation board to the host system.
3. Install the necessary driver software for the LILY-W1 series modules as described in section 4.

Figure 4: J2 and J3 jumper settings for SDIO

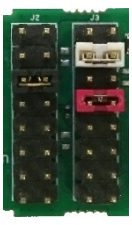


Figure 5: J2 and J3 jumper settings for USB

2.1 Host interface through SDIO

The default jumper settings, as shown in Figure 4 and described in Table 4, enable SDIO as the host interface. Secure Digital (SD) card is used for power supply of the main and I/O voltages.

Jumper configuration	Description
J2: 7-8 bridged	SDIO interface selection
J3: 5-6 bridged	Selects 3.3 V SD card supply for main power supply
J3: 3-4 bridged	Selects 3.3 V I/O voltage

Table 4: Jumper configuration for SDIO interface

Connect the evaluation board to an SDIO capable host by inserting the board into the SD card slot. As the LILY-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" *[4]*.

⚠ Be careful while inserting the evaluation board 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.

2.2 Host interface through USB

The jumper settings for selecting the USB interface for host communication are shown in Figure 5 and described in Table 5. Use a USB micro-B cable to connect the evaluation board to the host through the micro-USB receptacle on the evaluation board.

Jumper configuration	Description
J2: 5-6 bridged	USB interface selection. SD_CMD/USB_VBUS_ON connected to VBUS (via 33k and 3.3V LDO)
J3: 7-8 bridged	Selects USB supply for main power supply
J3: 3-4 bridged	Selects 3.3 V I/O voltage

Table 5: Jumper configuration for USB interface

3 Board description

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

3.1 Connectors

Table 6 lists the available connectors on the EVB-LILY-W1 evaluation board and their functions. Refer to the schematic in section 3.6 for details about the pin assignment.

Function	Description	Name
SDIO interface	SD card connector for connecting to the host SDIO socket	J1
USB interface	Micro-USB 2.0 receptacle for connecting to the USB host	J7
Host interface selection / SDIO	Pin header for selecting the host interface and accessing the SDIO signals	J2
Power supply configuration / LILY-W1 module I/O	Pin header for power supply and I/O voltage selection and access to the I/O signals of the module	J3
Single external antenna connector	U.FL coaxial connector for single external 2.4 GHz Wi-Fi antenna (can only be used on the EVB-LILY-W131)	J6
Diversity external antenna connector 1	U.FL coaxial connector for first external 2.4 GHz Wi-Fi antenna for antenna diversity (can only be used on the EVB-LILY-W131); disconnected by default	J4
Diversity external antenna connector 2	U.FL coaxial connector for second external 2.4 GHz Wi-Fi antenna for antenna diversity (can only be used on the EVB-LILY-W131); disconnected by default	J5

Table 6: Description of available connectors on the EVB-LILY-W1

3.2 Pin header J2

The dual-row pin header J2 on the evaluation board is used to:

- Select between the SDIO and USB host interfaces
- Access the SDIO signals

Table 7 and Table 8 list the available host interface selection options and signals on J2.

Jumper configuration	Description
7-8 bridged, 5-6 open	Selected SDIO host interface (J1)
5-6 bridged, 7-8 open	Selected USB host interface (J7)

Table 7: J2 host interface selection

Pin	Signal	LILY-W1 signal	Description
1	SDCARD.D2	SDIO_D2	SDIO bidirectional data signal 2
3	SDCARD.D3	SDIO_D3	SDIO bidirectional data signal 3
5	SDCARD.CMD	SDIO_CMD/USB_VBUS_ON	SDIO: command/response signal USB: power valid indication
6	USB_3V3 via 33k	-	VBUS via 33 k and 3.3 V LDO
7	SDIO_EN-N	USB/SDIO-n	Host interface selection
9	SDCARD.VDD	-	SDIO 3.3 V power supply
11	SDCARD.CLK	SDIO_CLK	SDIO host to card clock signal
13	SDCARD.D0	SDIO_D0	SDIO bidirectional data signal 0
15	SDCARD.D1	SDIO_D1	SDIO bidirectional data signal 1
2, 4, 8, 10, 12, 14, 16	GND	GND	Ground

Table 8: J2 pin description

3.3 Pin header J3

The dual-row pin header J3 on the evaluation board is used to:

- Select the power supply and I/O voltages
- Provide access to the I/O signals of the LILY-W1 module

Table 9 and Table 10 list the available power supply configuration options and signals on J3.

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

Jumper configuration	Description
5-6 bridged, 7-8 open	3.3 V SDIO card supply is used for main power supply
7-8 bridged, 5-6 open	5 V USB supply is used to generate the 3.3 V main power supply
3-4 bridged, 1-2 open	3.3 V I/O voltage selected
1-2 bridged, 3-4 open	1.8 V I/O voltage selected, generated from the 3.3 V main supply

Table 9: J3 power supply configuration

Pin	Signal	LILY-W1 signal	Description
1, 3	VDDIO	VCC_IO	I/O voltage supply
2	V1_8V	-	1.8 V supply from internal LDO
4, 6, 8	VCC	VCC	Module main voltage supply
5	SDCARD.VDD	-	SDIO 3.3 V power supply
7	USB_3V3	-	3.3 V supply from VBUS and internal LDO
9	POWER_DOWN-N	PD-n	Power down, active low
11	ANT.SEL	ANT_SEL	External antenna selection signal for diversity (only on EVB-LILY-W131)
12	ANT.SEL-N	ANT_SEL-n	External inverted antenna selection signal for diversity (only on EVB-LILY-W131)
13	HOST_WKUP	HOST_WKUP	Module-to-Host Wake-Up signal
14	RESERVED	-	Reserved
15	WAKE_UP	WAKE_UP	Host-to-Module Wake-Up signal
10, 16	GND	GND	Ground

Table 10: J3 pin description

3.4 Antenna interfaces on the EVB-LILY-W131

The EVB-LILY-W131 evaluation board, which is used to evaluate the LILY-W131 module version with antenna pin, has three U.FL coaxial connectors J4, J5, and J6 for connecting the external antennas. The J6 connector is used for single antenna configuration and J4/J5 can be used for evaluating antenna diversity after switching components on the board.

3.4.1 Single antenna

This is the default configuration, where the U.FL coaxial connector J6 is connected to the antenna pin of the LILY-W131 module. Connect the included 2.4 GHz antenna to J6.

3.4.2 Antenna diversity

Antenna diversity can be evaluated using the U.FL coaxial connectors J4 and J5, after populating C12 instead of C13 on the evaluation board[1]. Connect two external 2.4 GHz antennas to J4 and J5. Antenna selection for diversity is controlled by the ANT.SEL and ANT.SEL-N signals from the LILY-W131 module. These signals can be monitored on the pin header J3 (see Table 10).

3.4.3 LTE filter

The LILY-W131 module includes a standard SAW bandpass filter. A BAW filter for Wi-Fi and LTE co-existence (Triquint 885071) is included on the evaluation board in the antenna path to the J5 U.FL connector by default (FL1), when using the antenna diversity configuration. The filter can be disabled by populating C18 and C20 instead of C19 and C21. For evaluating the performance of the LTE filter, the signals ANT.SEL and ANT.SEL-N on the pin header J3 can be tied to fixed values to manually select one antenna path. R7 and R8 must be removed first to disconnect the switching control from the LILY-W131 module. Table 11 lists the possible signal states for selecting one of the antenna paths.

ANT.SEL	ANT.SEL-N	Selected antenna connector
GND	VCC	J5 (with LTE filter)
VCC	GND	J4 (without LTE filter)

Table 11: Antenna selection

3.5 LED

Table 12 lists the available LED on the EVB-LILY-W1 evaluation board:

Function	Description	Name	Color
Main power	Main power supply (VCC) status indication	LD1	Green

Table 12: LED description

[1] Refer to the schematics in section 3.6 and assembly drawing in section 3.7 to find the location of the components.

3.6 Schematic

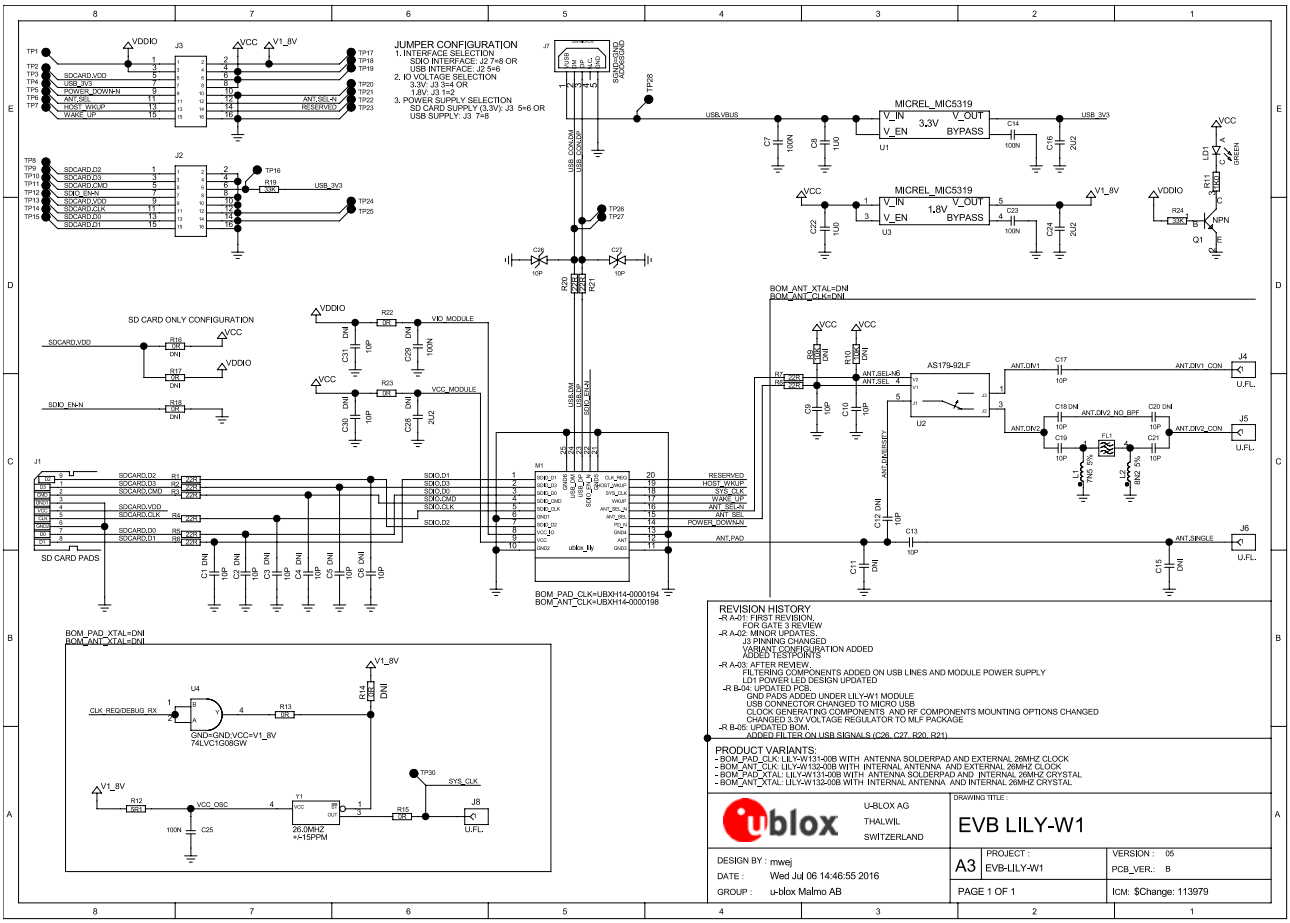


Figure 6: Schematic of the EVB-LILY-W1 evaluation board

3.7 Assembly

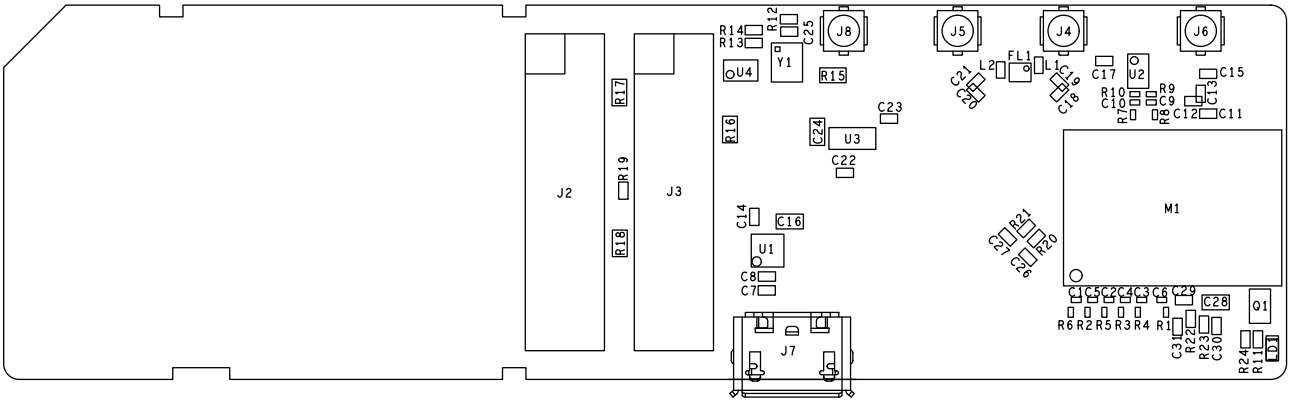


Figure 7: Assembly of the EVB-LILY-W1 evaluation board

4 Software

The LILY-W1 module series is based on the Marvell Avastar 88W8801 chipset and it supports Wi-Fi 802.11b/g/n simultaneous client/station, access point, and Wi-Fi Direct operations.

The LILY-W1 modules connect to the host processor either through an SDIO 2.0 or USB 2.0 device interface.

From the software point of view, the LILY-W1 series modules contain only calibration data and basic operation settings in an on-board non-volatile memory 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
- A driver, which is placed between the bus driver and the attached network stacks

Various control tools are also included optionally.

4.1 Driver versions

Marvell reference drivers for the LILY-W1 series modules are currently available for Linux and Android operating systems as SDIO or USB driver versions. The drivers are usually released for a single reference host platform and operating system version, but can be easily ported to comparable platforms. It is recommended to use the latest available host interface driver version and port this to the used operating system version.

⚠ The Software section of this manual describes only the Marvell reference drivers, which can be obtained through u-blox support. The “mwifiex” open source drivers that are distributed with the Linux kernel are not officially supported by u-blox.

☞ Refer to the Release Notes that is bundled with each driver release for a list of supported driver features.

4.2 Driver and firmware architecture

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

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

The host drivers interface with the SDIO or USB bus drivers and upper layer protocol stacks of the Linux/Android system. The basic architecture of the Wi-Fi driver is typical of a thick firmware architecture, where the Wi-Fi firmware handles all 802.11 MAC management tasks. Figure 8 shows the basic driver and firmware architecture.

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

- The driver registers itself with the MMC/SDIO or USB 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.

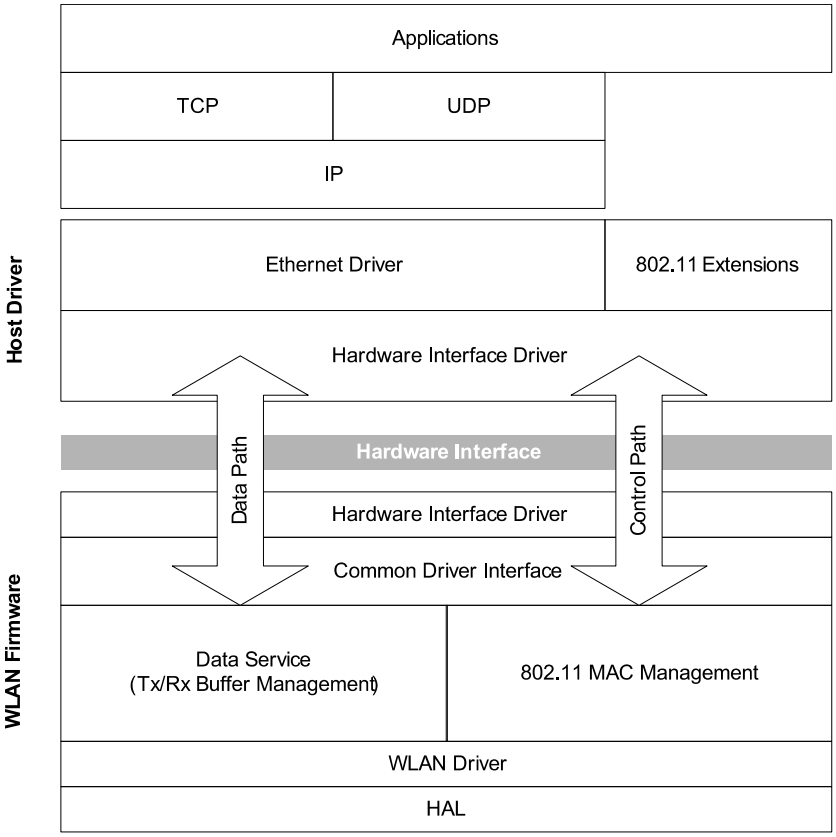


Figure 8: Basic Wi-Fi driver and firmware architecture

4.3 Compiling the drivers

4.3.1 Prerequisites

4.3.1.1 Reference drivers

The versions of the Marvell reference drivers/firmware package and the Linux OS that are described in the Software section of this document are:

- Marvell Linux reference driver:
  SD-UAPSTA-8801-FC18-MMC-14.85.36.p101-C3X14160_B0-GPL
- Linux kernel 3.19.8

☞ The instructions for compilation and usage of the USB driver version are very similar to the SDIO version. The main differences between both versions are noted in the description, if applicable.

The drivers should be able to support Linux kernel versions from 2.6.13 to 4.2. Older or more recent kernels might require some patches due to the changed kernel APIs. Patches for 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 LILY-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

According to the used host interface, the drivers for the LILY-W1 series modules depend on the MMC/SDIO or the USB stack of the Linux kernel; thus the respective stack 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 the package content

The Marvell driver package contains the firmware image, the Wi-Fi 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-8801-FC18-MMC-14.85.36.p101-C3X14160_B0-GPL-Release.zip
tar xf SD-UAPSTA-8801-FC18-MMC-14.85.36.p101-C3X14160_B0-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/
│   └── sd8801_uapsta.bin         # SDIO firmware image (USB version: usb8801_uapsta.bin)
└── SD-8801-FC18-MMC-14.85.36.p101-C3X14160_B0-GPL/
    └── wlan_src/                 # Wi-Fi driver and tools sources
        ├── Makefile              # Driver/tools Makefile
        ├── mapp/                 # User space tools for configuration, sample configuration 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 the test modes for certification *[5].*

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 the module versioning work.

4.3.4.2 Wi-Fi driver and tools

To compile the Wi-Fi drivers and tools, go to the driver packages's `wlan_src` subdirectory 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=~/elin-w160-evk/build/tmp/sysroots/elin-w160-evk/usr/src/kernel/

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_sd8801/` (or `../bin_usb8801` in case of USB driver), related to the `wlan_src` directory. The following table summarizes the content of the Wi-Fi build results directory:

File	Description
mlan.ko, sd8801.ko/usb8801.ko	Wi-Fi driver kernel modules
README*	Usage instructions for the provided tools
config/*	Sample configuration files used by various tools
uaputl.exe	Micro-AP configuration tool
mlanutl	Configuration tool for additional driver parameters
mlanevent.exe	Netlink event listener
mlan2040coex	802.11 20/40 MHz coexistence handler

Table 13: Content of the Wi-Fi 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 `sd8801_uapsta.bin` (or `usb8801_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
│   │       └── sd8801_uapsta.bin
│   └── modules
│       └── 3.19.8
│           └── updates
│               ├── mlan.ko
│               └── sd8801.ko
└── opt
    └── lily-w1
        └── bin_sd8801
```

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
wpa_supplicant	WPA supplicant. Handles key negotiation and roaming etc., on the 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 the regulatory domain

Table 14: Recommended additional software packages

4.5 Loading the drivers

4.5.1 SDIO driver

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

```
modprobe sd8801 cfg80211_wext=0xf
```

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

This will automatically load the *sd8801* 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 sd8801`" command. If the drivers are successfully loaded, you should see them in the list of loaded modules as shown below (via `lsmod` command):

☞ The internal name for the sd8801 module is sd8xxx.

```
Module                  Size  Used by
sd8xxx                357725  0
mlan                  251382  1 sd8xxx
```

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.

```
wlan: Loading MWLAN driver
wlan: Driver loaded successfully
mmc1: new high speed SDIO card at address 0001
vendor=0x02DF device=0x9139 class=0 function=1
rx_work=0 cpu_num=1
Request firmware: mrvl/sd8801_uapsta_sdio.bin
Wlan: FW download over, firmwarelen=234524 downloaded 234524
WLAN FW is active
fw_cap_info=0xba3, dev_cap_mask=0xffffffff
wlan: version = SD8801-14.85.36.p101-C3X14C160-GPL-(FP85)
```

Listing 5: 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

Table 15: Network interfaces

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

```
$ mlanutl mlan0 version
Version string received: SD8801-14.85.36.p101-C3X14C160-GPL-(FP85)
$ iwpriv mlan0 version
mlan0     version:SD8801-14.85.36.p101-C3X14C160-GPL-(FP85)
```

4.5.2 USB driver

The USB driver can be loaded by issuing the following command, which will load the *usb8801* kernel module and all its dependencies such as *mlan* or *cfg80211*:

```
modprobe usb8801 cfg80211_wext=0xf
```

For a full description of the available module parameters refer to the README files and also to the output of the "`modinfo usb8801`" command. If the drivers are successfully loaded, you should see them in the list of loaded modules as shown below (via `lsmod` command):

☞ The internal name for the usb8801 module is usb8xxx.

```
Module                  Size  Used by
usb8xxx               349149  0
mlan                  243114  1 usb8xxx
```

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

```
wlan: Loading MWLAN driver
usbcore: registered new interface driver usb8xxx
wlan: Driver loaded successfully
usb 2-1: new high-speed USB device number 4 using musb-hdrc
[…]
Request firmware: mrvl/usb8801_uapsta_usb.bin
WLAN FW is downloaded
usb_reset_device() successful.
usb 2-1: USB disconnect, device number 4
usb 2-1: new high-speed USB device number 5 using musb-hdrc
sb 2-1: New USB device found, idVendor=1286, idProduct=204a
usb 2-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
usb 2-1: Product: Marvell Wireless Device
usb 2-1: Manufacturer: Marvell
usb 2-1: SerialNumber: D4CA6E9001F7
VID/PID = 1286/204A, Boot2 version = 3117
rx_work=0 cpu_num=1
WLAN FW is active
fw_cap_info=0xba3, dev_cap_mask=0xffffffff
wlan: version = USB8801-14.85.36.p101-C3X14C160-GPL-(FP85)
```

Listing 6: Kernel log after connecting via USB

After successful registration, the network interfaces listed in Table 15 should be available. The version of the loaded firmware can be verified for example, by using one of the following commands:

```
$ mlanutl mlan0 version
Version string received: USB8801-14.85.36.p101-C3X14C160-GPL-(FP85)
$ iwpriv mlan0 version
mlan0     version: USB8801-14.85.36.p101-C3X14C160-GPL-(FP85)
```

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                  # for SDIO driver; or
rmmod mlan usb8xx                  # for USB driver
```

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 LILY-W1                          # set AP SSID to "LILY-W1"
./uaputl.exe sys_cfg_channel 6                             # set AP radio channel to 6
./uaputl.exe sys_cfg_11n 1 0x112c 3 0 0xff                 # enable 802.11n mode with short guard interval
./uaputl.exe sys_cfg_80211d state 1 country US             # enable 802.11d, set country

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

./uaputl.exe bss_start                                     # start the AP
```

Listing 7: Create a Wi-Fi access point

⚠ The capability of the provided reference tools (for example, uaputl) differs depending on the driver package versions.

For older versions of the driver package, use the following commands to set the AP configuration including SSID, channel and encryption, and to start the AP:

```
iwpriv uap0 apcfg "ASCII_CMD=AP_CFG,SSID=LILY-W1,CHANNEL=6,SEC=wpa2-psk,KEY=topsecret"
iwpriv uap0 bsstart
```

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.

The list of client stations that are currently associated with the AP can be obtained with the "`uaputl sta_list`" command, for example:

```
./uaputl sta_list
Number of STA = 1

STA 1 information:
====================
MAC Address: 60:67:20:xx:xx:xx
Power mfg status: active
Rssi : -56 dBm
```

4.6.2 Wi-Fi station mode

4.6.2.1 Using Marvell tools

This example will connect the LILY-W1 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
```

The "`iwconfig mlan0`" command can be used to display parameters and statistics of the wireless network interface.

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 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

Abbreviation	Definition
AP	Access point
API	Application Programming Interface
CLI	Command-Line Interface
DHCP	Dynamic Host Configuration Protocol
EVB	Evaluation Board
EVK	Evaluation Kit
GND	Ground
GPIO	General-purpose input/output
I/O	Input/Output
IP	Internet Protocol
LDO	Low Drop Out
LED	Light-Emitting Diode
LTE	Long Term Evolution
MAC	Medium Access Control
MMC	MultiMedia Card
OS	Operating System
SD	Secure Digital
SDIO	Secure Digital Input Output
SSID	Service Set Identifier
STA	Station
TCP	Transmission Control Protocol
uAP	Micro Access Point
UDP	User Datagram Protocol
USB	Universal Serial Bus
VCC	IC power-supply pin
WEXT	Wireless Extensions
WFD	Wi-Fi Direct
Wi-Fi	Wireless Local Area Network
WPA	Wi-Fi Protected Access

Table 16: Explanation of the abbreviations and terms used

Related documents

[1] u-blox Limited Use License Agreement for Marvell SW platform and OSS Deliverables
[2] LILY-W1 System Integration Manual, Document No. UBX-15027600
[3] LILY-W1 series Data sheet, Document No. UBX-15000203
[4] Sample card reader for Linux: Sonnet SDXC UHS-I Pro Reader/Writer ExpressCard/34 - http://www.sonnettech.com/product/sdxcproreader.html
[5] Radio Test Guide Application Note, Document No. UBX-15014433

☞ For regular updates to u-blox documentation and to receive product change notifications, register on our homepage (www.u-blox.com).

Revision history

Revision	Date	Name	Comments
R01	8-Mar-2016	mzes, kgom	Initial release.
R02	8-Apr-2016	mzes	Document status updated to Advance Information. Updated the version of the USB driver package and removed information about previously required patches. Added information about the part number of the LTE filter on EVK-LILY-W131 (section 3.4.3). Added reference to the Radio Test Guide Application Note.
R03	19-Aug-2016	mzes, kgom	Modified the document status to Early Production Information. Updated section 4 - Software to the latest SDIO and USB reference driver versions. Updated external antennas included in EVK-LILY-W131. Updated EVB schematics in section 3.6.
R04	3-Jul-2018	shoe, kgom	Updated section 4.6.1 to cover missing command support in uaputl for setting passphrase on some driver package releases.

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@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 80 405 092 00
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 WiFi Development Tools - 802.11 ***category:***

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

Other Similar products are found below :

YSAEWIFI-1 SKY65981-11EK1 QPF7221PCK-01 SIMSA915C-Cloud-DKL SIMSA433C-Cloud-DKL ISM43903-R48-EVB-E QPF4206BEVB01 RN-G2SDK SKY85734-11EK1 SKY85735-11EK1 MIKROE-2336 EVAL_PAN1760EMK 3210 ATWINC1500-XPRO 2471 DM990001 WRL-13711 2999 ATWILC3000-SHLD DFR0321 TEL0118 3213 DFR0489 SLWSTK-COEXBP WRL-13804 DEV-13907 UP-3GHAT-A20-0001 3405 TEL0078 2680 2702 2821 3606 3653 3654 4178 4285 CS-ANAVI-25 CS-ANAVI-26 CS-ANAVI-23 CS-ANAVI-24 CS-ANAVI-28 CS-ANAVI-29 CS-ANAVI-30 CS-ANAVI-31 ABX00021 ABX00023 AKX00004 AKX00014 AKX00015