



PRM2141X Operating Instructions

Revision 2211.1

01 Nov 2022

Peraso Inc.
3905 Bering Dr. San Jose, CA
www.perasotech.com

Table of Contents

1. Introduction	3
2. Module Operation	4
2.1 RF Configuration.....	4
2.1.1 RF Parameters.....	4
2.1.2 RF Tuning	4
2.2 Driver Installation.....	4
2.2.1 Loading Driver	4
2.2.2 Removing Peraso driver	5
2.3 Driver Configuration.....	5
2.3.1 Setting Driver Parameters.....	5
2.3.2 Getting Wireless Interface Name.....	5
2.3.3 Setting Adapter IP Address	6
2.3.4 Setting up Access Point	6
2.3.5 Setting up Station	7
2.3.6 Connections for Secure Link	8
3. Configuration tools	10
3.1 Using vendor ID with the Adapter.....	10
3.2 Configuration utility prs_serial.....	10
3.2.1 Building prs_serial utility on Linux	10
3.3 Flash Utility.....	10
4. Appendix	11
4.1 Module Parameter Settings	11
4.2 Trace Features	12

1. Introduction

This manual provides the basic operating instructions for the PRM2141X-V-EGx (x = versions S, F and N). These instructions include the steps required for software driver initialization, tuning, and normal operation.

The PRM2141X is a complete USB 3.0 to WiGig module with advanced features for long range, outdoor applications. It utilizes the Peraso X720 802.11ad 60 GHz phased array chipset which includes a baseband processor and a high-power mmWave beamforming transceiver RFIC.

The PRM2141X incorporates a 16-element phased array antenna which can be used with or without a dish reflector. The antenna is integrated into the PCB and provides uniform performance over the entire unlicensed 802.11ad and 802.11ay “WiGig” band from 57 to 71 GHz.

2. Module Operation

The PRM2141X module is controlled by the software driver which must be installed on the host operating system. The module will be recognized and configured as a wireless adapter.

The module may be configured as an Access Point (AP) or as a Station (STA).

2.1 RF Configuration

2.1.1 RF Parameters

The only RF parameter that needs to be set during normal operation is the Channel for the defined SSID. The SSID name and channel are defined on the Access Point. STA devices will automatically scan channels to find the defined SSID for association.

See 2.3.4 and 2.3.5 to see command string required to set channel and SSID.

2.1.2 RF Tuning

The PRM2141X does not require any tuning or calibration for normal operation.

All RF maintenance functions, such as temperature compensation for power, LO leakage and frequency stability are done automatically by the module-embedded control and without host-level interaction.

2.2 Driver Installation

2.2.1 Loading Driver

The following outlines the steps for loading the Peraso driver module on a Linux operating system.

Before loading Peraso driver module, ensure that the wireless regulatory domain is configured and set to US:

```
sudo iw reg set US
```

note: The regulatory domain controls the allowed channels for a given country.

Load Peraso driver module provided as part of the software package deliverable:

```
sudo depmod -a
```

```
sudo modprobe prs_falcon <param_settings>
```

This command adds the loadable and persistent (i.e. the driver settings will not change when the device is removed or inserted) Peraso driver kernel module to the Linux kernel.

param_settings is optional and is in the form of <param_name>=<param_value>. Multiple parameters can be set by simply separating each setting with a space. See Section 5.1 in the Appendix for all the available module parameter settings.

Peraso Inc.
3905 Bering Dr. San Jose, CA
www.perasotech.com

note: A preferred method of configuring parameter settings is by modifying the parameters within the configuration file as shown in 3.2.1.

(OPTIONAL) Check that driver was successfully loaded:

```
lsmod | grep prs_falcon
```

(OPTIONAL) To get the driver logs, issue the following command:

```
dmesg -T
```

2.2.2 Removing Peraso driver

Similar to loading driver, modprobe can also be used to remove driver:

```
sudo modprobe -rv prs_falcon
```

note: Modules are not automatically removed when the Peraso wireless device is unplugged or reset.

2.3 Driver Configuration

2.3.1 Setting Driver Parameters

To set driver module parameters, edit (or create) the configuration file:

/etc/modprobe.d/prs_falcon.conf.

In this file, you can set module parameters by adding a line like the following:

```
options prs_falcon <parameter_name>=<parameter_value>
```

Here is an example of a commonly used module parameter:

```
options prs_falcon PRS_TRACE_FEATURES=538
```

Note: The driver must be removed (if it was already loaded), for the new settings to take effect.

2.3.2 Getting Wireless Interface Name

The interface_name can be obtained using the command:

```
ip addr
```

This will list all the interfaces and their relevant information.

Peraso Inc.
3905 Bering Dr. San Jose, CA
www.perasotech.com

Each wireless interface is denoted as wlanX (or wlX), where X is the interface number assigned by the operating system. The adapter with the MAC address of 20:ce:c4:##:##:##, where the last 6 hexadecimal digits are variable, is the Peraso wireless adapter.

2.3.3 Setting Adapter IP Address

An IP address can be configured for each node by using the following command:

```
ifconfig <interface_name> <ip_address> netmask <value> up
```

An example of setting the adapter IP address is:

```
sudo ifconfig wlan0 10.10.10.100 netmask 255.0.0.0 up
```

note: replace "wlan0" with the actual interface name (may be in the form wlx20cec4##### as mentioned above)

If setting up a link between two adapters, repeat the steps above for the second adapter, giving it a different <ip_address> (eg 10.10.10.101)

2.3.4 Setting up Access Point

The service **hostapd** is used to configure the role of an Access Point (AP). This can be achieved by setting up a configuration file that the hostapd will invoke using the parameters in the file.

The following example shows a hostapd configuration file 'hostapd.conf' for a non-secure (open) link named 'clean-link'

```
#example content of hostapd.conf file
driver=nl80211
interface=wlan0
ctrl_interface=/var/run/hostapd
ssid=clear-link
channel=2
hw_mode=ad
country_code=US
ctrl_interface_group=0
```

note: replace "wlan0" with the actual interface name (may be in the form wlx20cec4##### as mentioned above)

After creating the hostapd.conf file, hostapd is launched by using the following command from the same directory:

```
sudo hostapd -ddddd -g /var/run/hostapd/global hostapd.conf
```

Peraso Inc.
3905 Bering Dr. San Jose, CA
www.perasotech.com

At this point the adapter should be sending beacons and waiting for a connection. You may see a bunch of output to the terminal from time to time. To further interact with the adapter (eg to check or set MIBs using `prn_serial`) and/or network (eg to run traffic using `iperf` or to ping) it is suggested to open up more terminals on the host.

2.3.5 Setting up Station

The service **wpa_supplicant** is used to configure the role of a Station (STA). This can be achieved by setting up a configuration file that the `wpa_supplicant` will invoke using the parameters in the file.

The following example shows a `wpa_supplicant` file 'wpa.conf' for a non-secure (open) link named 'clear-link'

```
#example content of wpa.conf file
ctrl_interface=/var/run/wpa_supplicant
network={
    ssid="clear-link"
    frequency=60480
    scan_ssid=1
    key_mgmt=NONE
}
```

After creating the `wpa.conf` file, `wpa_supplicant` is launched by using the following command from the same directory:

```
wpa_supplicant -D<driver> -i<interface> -c<conf file>
```

For example:

```
sudo wpa_supplicant -Dnl80211 -iwlan0 -c wpa.conf
```

note: replace "wlan0" with the actual interface name (may be in the form `wlx20cec4#####` as mentioned above)

At this point, the station should connect to the network based on the parameters in `hostapd.conf` and `wpa.conf` on the two hosts. You should be able to ping or run traffic from one host to the other using the IP addresses specified (eg 10.10.10.100 and 10.10.10.101 based on examples above).

note: to ensure that there is no conflict with a Wi-Fi network on the laptop, you should disable the Network Manager by issuing the following commands:

```
sudo systemctl stop NetworkManager.service
sudo systemctl disable NetworkManager.service
```

2.3.6 Connections for Secure Link

The procedure to establish secure link is similar to that of the non-secure link.

User creates hostapd and wpa_supplicant configure files with the secure information, then use the same procedure above to start the hostapd and wpa_supplicant.

The example below shows a pair of configuration for secure link.

The following example shows a hostapd configuration file 'hostapd-secure.conf' for a secure link named 'securelink'

#example content of hostapd-secure.conf file

```
driver=nl80211
interface=wlan4
ctrl_interface=/var/run/hostapd
ssid=secure-link
channel=2
hw_mode=ad
country_code=US
auth_algs=1
num_of_ptksa_replay_counters=3
num_of_gtksa_replay_counters=3
wpa=2
wpa_key_mgmt=WPA-PSK
wpa_pairwise=GCMP
wpa_group_rekey=0
wpa_passphrase=12345678
```

The following example shows a wpa_supplicant configuration file 'wpa-secure.conf' for a secure link named 'secure-link'

#example content of wpa-secure.conf file

```
ctrl_interface=/var/run/wpa_supplicant
num_of_ptksa_replay_counters=3
num_of_gtksa_replay_counters=3
network={
    ssid="secure-link"
    frequency=60480
    scan_ssid=1
    key_mgmt=WPA-PSK
```

```
proto=WPA2  
pairwise=GCMP  
group=GCMP  
psk="12345678"  
}
```

3. Configuration tools

3.1 Using vendor ID with the Adapter

Configure the Adapter from the user space would be using Vendor through the hostapd or wpa_supplicant. Please obtain document from Peraso for list of supported Vendor commands.

3.2 Configuration utility prs_serial

The prs_serial utility provides the ability to send CLI commands and receive a response. The CLI allows for low-level query and control of the MAC firmware.

3.2.1 Building prs_serial utility on Linux

The Peraso utility source is delivered as part of the official software release package. A user must compile and build the prs_serial executable. This can be achieved by executing the following command from the terminal.

```
# cd prs_serial/src
# make
```

If successful, the prs_serial executable can then be used to configure the supported CLI commands. The syntax for using the utility is shown below:

```
# ./prs_serial "<command> <argument1> <argument2>"
```

An example of a prs_serial to set MIB is shown below:

```
# ./prs_serial "mib q b 11-1-1-1"
```

Note if the preceding command does not return any output, try using sudo.

By default prs_serial talks to /dev/ttyACM0.

3.3 Flash Utility

The peraso_download_flashutility is used to burn the firmware image, one-time programming (OTP) and/or configuration database sections to an attached flash memory. The utility uses Falcon CLI protocol over a serial connection to write to the flash. Additional instructions are provided in a help menu text when running the utility.

Example of updating the image on the flash:

```
# prs_download_flash -i image.bin
```

Peraso Inc.
3905 Bering Dr. San Jose, CA
www.perasotech.com

4. Appendix

4.1 Module Parameter Settings

The table below outlines all the driver module parameters that can be configured via a configuration file or passed as argument to the modprobe command when loading driver.

Table 4-1: Peraso driver configuration parameters

Parameter	Descriptions	Default
PRS_DRIVER_MODE	The driver operation mode 0 – Normal operation	0
PRS_DEVICE_MODE	The Station Mode type: 0 – PCP 1 – STA 2 – Discovery	2
PRS_AMSDU_TYPE	Enables A-MSDU feature in the driver 0 – A-MSDU disabled 1 – Basic A-MSDU enabled 2 – Short A-MSDU enabled	0
PRS_MAX_AMSDU_SIZE_BYTES	Maximum AMSDU size in bytes	7935
PRS_ENABLED_CHANNELS_MASK	The enabled channels mask 0x1 – Channel 1 enabled 0x2 – Channel 2 enabled 0x4 – Channel 3 enabled 0x8 – Channel 4 enabled	0x2
PRS_ACTIVE_CHANNEL_SCAN_TIME_TU	The active channel scan time in TU	2000
PRS_PASSIVE_CHANNEL_SCAN_TIME_TU	The passive channel scan time in TU	2000
PRS_PROBE_DELAY_US	Delay in microseconds to be used by STA prior to transmitting a probe frame.	0
PRS_PBSS_SSID	The specific or broadcast SSID. Maximum of 32 bytes length.	PerasoWiGig
PRS_MAX_STA_TABLE_ENTRIES	Maximum number of station table entries maintained	4
PRS_MAX_STA_TABLE_LIFE_TIME_SEC	Maximum life time of station table entries.	29
PRS_ASSOC_FAILURE_TIMEOUT_US	Association failure timeout microseconds.	3000
PRS_TRACE_FEATURES	Bitmask of tracing features that can be enabled when debug compile enabled.	0
PRS_ENCRYPTION_MODE	Enables encryption feature	0

	0– Encryption disabled 1– Encryption enabled	
<i>PRS_MAX_LOST_BEACONS</i>	Specifies the maximum allowed lost beacons 0 – Device automatically chooses max lost beacons 1-32 – Valid values for the max lost beacons 255 – The maximum lost beacons is infinite – Feature disabled	0
<i>PRS_MAX_BSS_IDLE_PERIOD</i>	Specifies the maximum allowable idle period in ms 0 – The maximum idle period is infinite - Feature is disabled	3200
<i>PRS_TX_COMPLETE</i>	Enables TX status notification	0
<i>PRS_NETWORK_ADDRESS</i>	Mechanism to override device's MAC Address	
<i>PRS_ACK_TIMEOUT_US</i>	Configures the ACK Timeout in the system	8

4.2 Trace Features

Table 4-2 defines the bit masks for enabling the logging in the driver for the applicable functionalities.

Table 4-2: Bit mask for trace features in driver

Bit	Trace Feature Code	Default
0	FUNCTION	0
1	SCAN	0
2	STATION	0
3	CONNECT	0
4	MIB	0
5	TX_DATA	0
6	RX_DATA	0
7	PACKET	0
8	ETHER_SETUP	0
9	DRIVER_INIT	0
10	VIF	0

Bit	Trace Feature Code	Default
11	NOTIF	0
12	CONTROL	0
13	USB	0
14	URB	0
15	SIM	0
16	FLOW	0
17	STATS	0
18	MEM	0
19	NETLINK	0
20	TX-STATUS	0
21-31	Reserved	0