

**LK-RZG-A02****無線 LAN モジュール(WM-RP-10)の使用方法**

Rev1.0 2019/10/25

**目 次**

<b>1. 概要</b>	<b>3</b>
1.1 はじめに.....	3
1.2 開発環境.....	3
<b>2. Linux カーネルへの対応方法</b>	<b>4</b>
2.1 Linux カーネルへのドライバ追加.....	4
2.2 Linux カーネルの作成.....	7
<b>3. パラメータ設定方法</b>	<b>8</b>
3.1 WLAN.....	8
3.2 BT.....	10
3.3 BLE.....	11
<b>4. Bluetooth サンプルアプリのビルド</b>	<b>12</b>
<b>5. 動作確認</b>	<b>15</b>
5.1 Linux の起動.....	15
5.2 WLAN.....	15
5.3 BT.....	21
5.4 BTLE.....	23

## 表記

### ●バージョンに関する表記

弊社提供のソース等に関しては、弊社の管理するバージョン番号がファイル名やフォルダ名に付いている場合があります。そのバージョン番号に関しては、本ドキュメントでは、『X』を使用して表現しております。そのため、以下のような表記になりますので、その部分は読み替えてください。

例：

以下の表記がある場合

helloworld-X.X.tar.bz2

Ver1.0 での実際のファイル名は、以下になります。

helloworld-1.0.tar.bz2

### ●コマンドラインの表記

本ドキュメントには、コマンドラインで入力する操作手順が記載されております。操作は PC 及び XG ボードで行います。それぞれの記述について以下に記載します。

ゲスト OS(Ubuntu)での操作


プロンプトは、『\$』で記載します。

実際のプロンプトには、カレントディレクトリ等が表示されますが、本ドキュメントでは省略します。

AP-RZG-0A ボード上の Linux での操作

プロンプトは、『#』で記載します。


実際のプロンプトには、カレントディレクトリ等が表示されますが、本ドキュメントでは省略します。

本ドキュメント中での入力では、以下のように表現し、入力の最後には、 があります。

例：ゲスト OS(Ubuntu)上で make コマンドを実行する場合の表記

```
$ make 
```

コマンドによっては 1 つのコマンドが複数行で記載されている場合もあります。

その場合には、2 行目以降の入力では ENTER キーを押さずに続けて入力し、 の表記がある行の最後で ENTER キーを入力してそのコマンドを実行してください。

例：2 行続いてコマンド入力がある表記

```
$ cd ~/build/tmp/work/aprzg0a-poky-linux-gnueabi/linux-renesas/3.10+git34547b2a503c  
e6dca24b745d608d2f3baac187f-r0/git 
```

# 1. 概要

## 1.1 はじめに

本ドキュメントでは、AP-RZG-0A に無線 LAN モジュール/Bluetooth を接続して動作を行う方法を説明します。

- ・カーネル対応方法
- ・無線 LAN/Bluetooth 設定、動作確認

## 1.2 開発環境

本ドキュメントでは、Yocto/Poky 開発環境が Ubuntu にインストールされていることが前提となっています。

無線 LAN モジュールのドライバを Linux カーネルに組み込むには、カーネルのビルド環境が必要となります。カーネルのビルド環境設定は以下のドキュメントを参照してください。

- ・LK-RZG-A02 Install Manual
- ・LK-RZG-A02 Software Manual

## 2. Linux カーネルへの対応方法

Linux カーネルのデフォルトでは、無線 LAN モジュールを使用する設定になっておりませんので、Linux カーネルを再作成する必要があります。

その再作成する手順を説明します。



本手順では、開発環境がコマンド『`bitbake core-image-weston`』にて作成されていることを前提で説明します。  
コマンドに関する詳細は、AP-RZG-0A の Linux 開発キット (LK-RZG-A02) のソフトウェアマニュアルでご確認ください。

### 2.1 Linux カーネルへのドライバ追加

Linux カーネルのデフォルトは、無線 LAN モジュールを使用する設定になっておりませんので、以下の手順で Linux カーネルに無線 LAN モジュールのドライバを組み込むように変更します。

- ① ビルド環境の設定をします。

```
$ cd ~/user_work
$ source poky/oe-init-build-env
```

環境設定が終了すると、カレントディレクトリは~/user\_work/build に移動します。



カーネルのコンフィギュレーションを初期化する場合は、以下のコマンド実行します。

```
bitbake -c configure linux-renesas --force
```

このコマンドを実行すると以前に行われた menuconfig による設定変更、およびドライバなどのソースの変更は全て初期化されます。

- ② カーネルのカスタマイズをするため、設定画面を開きます。

```
$ bitbake -c menuconfig linux-renesas
```

```
.config - Linux/arm 4.4.138 Kernel Configuration
Linux/arm 4.4.138 Kernel Configuration
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty
submenus ----). Highlighted letters are hotkeys. Pressing <Y>
includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to
exit, <?> for Help, </> for Search. Legend: [*] built-in [ ]
[*] Patch physical to virtual translations at runtime
  General setup --->
[*] Enable loadable module support --->
[*] Enable the block layer --->
  System Type --->
  Bus support --->
  Kernel Features --->
  Boot options --->
  CPU Power Management --->
  Floating point emulation --->
<Select> <Exit> <Help> <Save> <Load>
```

- ③ Wireless LAN メニューに移動します。

[Device Drivers] - [Network device support] - [Wireless LAN]の順に開いていき、『WM-RP-10』を選択します。

```
.config - Linux/arm 4.4.138 Kernel Configuration
> Device Drivers > Network device support > Wireless LAN
Wireless LAN
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty
submenus ----). Highlighted letters are hotkeys. Pressing <Y>
includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to
exit, <?> for Help, </> for Search. Legend: [*] built-in [ ]
< > IEEE 802.11 for Host AP (Prism2/2.5/3 and WEP/TKIP/CCMP)
< > Intel PRO/Wireless 2100 Network Connection (NEW)
< > Intel PRO/Wireless 2200BG and 2915ABG Network Connection (N
< > Marvell 8xxx Libertas WLAN driver support (NEW)
< > Hermes chipset 802.11b support (Orinoco/Prism2/Symbol) (NEW
[ ] Mediatek Wireless LAN support ----
[ ] TI Wireless LAN support ----
< > Marvell WiFi-Ex Driver (NEW)
[ ] WM-RP-04S/05S
[*] WM-RP-10
<Select> < Exit > < Help > < Save > < Load >
```



上記の画面で何も項目が表示されない場合は、一つ上の階層の『Wireless LAN』がチェックされているかご確認ください。

- ④ ESCキーを複数回押し、トップメニューに戻り、設定値を保存します。

```
.config - Linux/arm 4.4.138 Kernel Configuration

Do you wish to save your new configuration?
(Press <ESC><ESC> to continue kernel configuration.)
< Yes > < No >
```

- ⑤ デバイスツリーファイルを編集します。  
以下に編集ファイル名、ファイル場所、編集内容を記載します。

●ファイル名

r8a7745-aprzg0a.dts

●ファイル場所

```

/
|-- home
  |-- guest
    |-- user_work
      |-- build
        |-- tmp
          |-- work-shared
            |-- aprzg0a
              |-- kernel-source
                |-- arch
                  |-- arm
                    |-- boot
                      |-- dts

```

●編集内容

ファイル『r8a7745-aprzg0a.dts』の21行目から各無線LANモジュールの#define定義が記述されています。

以下のようにWMRP\_10の定義のみコメント解除されるように変更します。

```

/* WM-RP */
//#define WMRP_0xS
#define WMRP_10

```

## 2.2 Linux カーネルの作成

Linux カーネルのみ再ビルドする手順を説明します。

- ① カーネルのビルドをします。

```
$ bitbake -c compile linux-renesas --force
Loading cache: 100% |#####| Time: 0:00:00
Loaded 3036 entries from dependency cache.
NOTE: Resolving any missing task queue dependencies

:
<途中省略>
:

NOTE: Executing RunQueue Tasks
NOTE: Tasks Summary: Attempted 334 tasks of which 331 didn't need to be rerun and all succeeded.

Summary: There were 2 WARNING messages shown.
```

- ② カーネルのビルドが成功したら、作成されたカーネルをデプロイ(配布)します。

```
$ bitbake -c deploy linux-renesas
Loading cache: 100% |#####| Time: 0:00:00
Loaded 3036 entries from dependency cache.
NOTE: Resolving any missing task queue dependencies

:
<途中省略>
:


NOTE: Executing RunQueue Tasks
NOTE: Tasks Summary: Attempted 473 tasks of which 462 didn't need to be rerun and all succeeded.

Summary: There were 2 WARNING messages shown.
```

- ③ 正常に完了しますと、ディレクトリ『./tmp/deploy/images/aprzg0a』にファイルが生成されます。無線LANモジュールの追加で更新するファイルは、以下となります。

ファイル名	内容
uImage	カーネルイメージファイル
r8a7745-aprzg0a.dtb	デバイスツリーファイル

Table 2.3-1 更新ファイル

 作成された Linux カーネルを microSD カードにコピーして動作できるようにする必要があります。その手順に関しては、AP-RZG-0A の Linux 開発キット (LK-RZG-A02) のソフトウェアマニュアルでご確認ください。

## 3. パラメータ設定方法

本章では、無線 LAN モジュールを動作させるにあたり、パラメータ設定の変更方法を説明します。

### 3.1 WLAN

#### 3.1.1 iwconfig

「iwconfig」のパラメータで割りあっている特殊な機能について説明します。

コマンド	設定値	詳細
Mode	“Managed”	インフラストラクチャモードで動作
	“Master”	アクセスポイントモードで動作
Channel (※1)	Infrastructure : 0~13, 36~165	動作チャンネルの設定 「0」を指定した場合は Dual band モードで動作します。
	Master : 1~13, 36~48	

Table3.1.1 iwconfig で設定可能な特殊パラメータ

※1 使用するリージョンによって使用可能なチャンネルが異なります。  
詳細は WM-RP-10 のハードウェアマニュアルをご確認ください。



### 3.1.2 iwpriv

「iwpriv」を使用したパラメータ設定方法を説明します。

コマンド	機能	設定値	詳細
version	ファームウェアバージョンの取得		
macaddr	MAC アドレスの取得		
securitymode	セキュリティモードの設定	0	Open(デフォルト)
		1	WPA-TKIP
		2	WPA2-TKIP
		3	WEP
		4	WPA-EAP
		5	WPA2-EAP
		6	WPA/WPA2 MIX
		7	WPA-CCMP
		8	WPA2-CCMP
region	リージョンの設定	0	デフォルト(US)
		1	US
		2	EU
		3	JP
antenna	使用するアンテナの設定	0	オンボードアンテナを使用 (デフォルト)
		1	外付けアンテナを使用
hidden	Hidden Access Point 機能の設定	0	Hidden AP 無効 (デフォルト)
		1	Hidden AP 有効
wpsmode	WPS 動作モードの設定	0	WPS 無効 (デフォルト)
		1	Push Button 方式
		2	PIN 方式(PIN 手動入力)
		3	PIN 方式(PIN 自動生成)
wps_start	WPS 動作の開始		
wpspin	WPS PIN コードの設定	任意の文字列 (32 文字)	
eapmethod	EAP 認証タイプの設定	任意の文字列 (16 文字)	
eapinner	EAP 認証方法の設定	任意の文字列 (64 文字)	
username	EAP ユーザ名の設定	任意の文字列 (32 文字)	
password	EAP パスワードの設定	任意の文字列 (32 文字)	

**Table3.1.2 iwpriv で設定可能なパラメータ**

※使用できる設定に関しては、WM-RP-10 のハードウェアマニュアル等をご確認ください。

※iwpriv (set)コマンドで指定されなかったパラメータについては、デフォルト値で動作します。

※文字列が返ってくる iwpriv (get)コマンドでは、文字列が格納されたアドレスが返されます。

## 3.2 BT

「ioctl」に割り当てられた機能と API を説明します。

コマンド	機能
WMRP10_IOCTL_APP_INIT	WMRP10 の初期化・リセット
WMRP10_IOCTL_APP_EXIT	WMRP10 の動作停止
WMRP10_IOCTL_BT_EVENT	Event 情報の取得(BT 用)

**Table3.2.1 ioctl で使用可能な機能**

コマンド	使用する API
WMRP10_IOCTL_BT_OPER_MODE (※1)	rsi_bt_oper_mode()
WMRP10_IOCTL_BT_SET_LOCAL_NAME	rsi_bt_set_local_name_mode()
WMRP10_IOCTL_BT_SET_PROFILE_MODE	rsi_bt_set_profile_mode()
WMRP10_IOCTL_BT_SET_DISCOVERY_MODE	rsi_bt_set_discovery_mode()
WMRP10_IOCTL_BT_SET_CONNECTABILITY_MODE	rsi_bt_set_connectability_mode()
WMRP10_IOCTL_BT_INQUIRY	rsi_bt_inquiry()
WMRP10_IOCTL_BT_INQUIRY_CANCEL	rsi_bt_inquiry_cancel()
WMRP10_IOCTL_BT_BOND	rsi_bt_bond()
WMRP10_IOCTL_BT_PINCODE_REPLY	rsi_bt_pincode_reply()
WMRP10_IOCTL_BT_LINKKEY_REPLY	rsi_bt_linkkey_reply
WMRP10_IOCTL_BT_SPP_CONNECT	rsi_bt_spp_connect()
WMRP10_IOCTL_BT_SPP_DISCONNECT	rsi_bt_spp_disconnect()
WMRP10_IOCTL_BT_TRANSFER	rsi_bt_spp_transfer()

**Table3.2.2 ioctl で使用可能な API(BT)**

※1 wlan 動作開始前に rsi\_bt\_oper\_mode() を実行することで、BT モードまたは BLE モードと WLAN モードを同時に動作させることができます。

rsi\_bt\_oper\_mode() の設定値によって動作させるモードが異なります。

設定値と動作モードの詳細は、WM-RP-10 のハードウェアマニュアル等をご確認ください

### 3.3 BLE

「ioctl」に割り当てられた機能とAPIを説明します。

コマンド	機能
WMRP10_IOCTL_APP_INIT	WMRP10の初期化・リセット
WMRP10_IOCTL_APP_EXIT	WMRP10の動作停止
WMRP10_IOCTL_BLE_EVENT	Event情報の取得(BLE用)

**Table3.3.1 ioctlで使用可能な機能**

コマンド	使用するAPI
WMRP10_IOCTL_BLE_SET_RANDOM_ADDRESS	rsi_bt_ble_set_random_address()
WMRP10_IOCTL_BLE_SCAN	rsi_ble_scan()
WMRP10_IOCTL_BLE_CONNECT	rsi_ble_connect()
WMRP10_IOCTL_BLE_DISCONNECT	rsi_ble_disconnect()
WMRP10_IOCTL_BLE_QUERY_DEVICE_STATE	rsi_ble_query_device_state()
WMRP10_IOCTL_BLE_SET_ADVERTISE_DATA	rsi_ble_set_advertise_data()
WMRP10_IOCTL_BLE_ADVERTISE	rsi_ble_advertise()
WMRP10_IOCTL_BLE_DEVICE_ADDSERVICE	rsi_ble_device_AddService()
WMRP10_IOCTL_BLE_DEVICE_ADDSERVICEATTRIBUTE	rsi_ble_device_AddServiceAttribute()
WMRP10_IOCTL_BLE_DEVICE_CHANGELOCALATTVALUE	rsi_ble_device_ChangeLocalAttValue()
WMRP10_IOCTL_BLE_QUERY_PROFILE	rsi_ble_query_profile()
WMRP10_IOCTL_BLE_READ_CHAR_VALUE_BY_UUID	rsi_ble_read_char_value_by_UUID

**Table3.3.2 ioctlで使用可能なAPI(BLE)**

## 4. Bluetooth サンプルアプリのビルド

WM-RP-10 の Bluetooth 動作確認を行うためのサンプルアプリのビルド方法を説明します。

### 作成のための準備

- ① 作業用ディレクトリ『**aprzg0a-app**』をホームディレクトリに作成します。  
すでに作成されている場合は、手順②にお進みください。

```
$ mkdir ~/aprzg0a-app
```

- ② ディレクトリ『**aprzg0a-app**』に移動します。

```
$ cd ~/aprzg0a-app
```

- ③ 作業用ディレクトリに付属 DVD-ROM 内の以下の 1 つのファイルをコピーします。  
手順④～⑥で例として DVD-ROM から直接コピーする方法を記述します。他の方法でコピーする場合には、コピー作業完了後に、手順⑦にお進みください。

wmrp10-X.X.tar.bz2

※『X.X』にはバージョン番号が入ります。Ver1.0 の場合は、『1.0』

- ④ DVD-ROM をドライブに挿入します。

デフォルトでは、自動でマウントされますが、マウントされない場合は、以下のコマンドを実行します。

```
$ gvfs-mount -d /dev/sr0
```



マウントされているかどうかは、『**mount**』コマンドで確認できます。  
以下のように、『**/dev/sr0**』が表示されている場合は、すでにマウントされています。  
(『\*\*\*\*\*』は、DVD-ROM のボリュームラベルになります。)

```
$ mount
:
<途中省略>
:
/dev/sr0 on /media/***** type udf (ro, nosuid, nodev, uhe lper=udisks, uid=1000,
gid=1000, iocharset=utf8, umask=0077)
```

- ⑤ 1 つのファイルをコピーします。コマンド途中の『\*\*\*\*\*』は、DVD-ROM のボリュームラベルになります。  
そのため、その部分は挿入した DVD-ROM に合わせて入力してください。

```
$ cp /media/guest/*****/sample/wmrp10-X.X.tar.bz2 .
```

- ⑥ DVD-ROM をアンマウントします。

```
$ umount /dev/sr0
```

- ⑦ サンプルソースを展開します。

```
$ tar -xjpf wmrp10-X.X.tar.bz2
```

## サンプルアプリケーションのビルド

サンプルアプリケーションのビルド手順を説明します。

- ① SDKの環境を設定します。

```
$ ./opt/poky/2.4.2/environment-setup-cortexa7hf-neon-poky-linux-gnueabi
```



最初の『.』と『/opt/...』の間には、半角スペースが必要ですので、ご注意ください。

『./opt/...』の環境設定コマンドは、各ターミナル毎に行う必要があります。

同一のターミナルで2回実行する必要はありませんが、別のターミナルを起動した場合には、再度実行する必要があります。

- ② 準備作業で展開した作業用ディレクトリの『wmp10\_bt\_sample』へ移動します。

```
$ cd ~/aprzg0a-app/wmp10_app/wmp10_bt_sample
```

- ③ サンプルアプリケーションをビルドします。

```
$ make
arm-poky-linux-gnueabi-gcc -march=armv7ve -mfpu=neon -mfloat-abi=hard -mcpu=cortex-a7
--sysroot=/opt/poky/2.4.2/sysroots/cortexa7hf-neon-poky-linux-gnueabi -I/home/guest/us
er_work/build/tmp/work-shared/aprzg0a/kernel-source/drivers/net/wireless
:
<途中省略>
:
-I/home/guest/user_work/build/tmp/work-shared/aprzg0a/kernel-source/drivers/net/wirele
ss/wmp10/wmp10_apis/apis/ble/core/include -DRSI_HAL -WI,-O1 -WI,--hash-style=gnu -W
l,--as-needed wmp10_bt_sample.c -o wmp10_bt_sample
```

ビルドが成功しますと、『wmp10\_bt\_sample』が作成されます。

- ④ 準備作業で展開した作業用ディレクトリの『wmp10\_ble\_sample』へ移動します。

```
$ cd ~/aprzg0a-app/wmp10_app/wmp10_ble_sample
```

- ⑤ サンプルアプリケーションをビルドします。

```
$ make
arm-poky-linux-gnueabi-gcc -march=armv7ve -mfpu=neon -mfloat-abi=hard -mcpu=cortex-a7
--sysroot=/opt/poky/2.4.2/sysroots/cortexa7hf-neon-poky-linux-gnueabi -I/home/guest/us
er_work/build/tmp/work-shared/aprzg0a/kernel-source/drivers/net/wireless/wmp10
:
<途中省略>
:
-I/home/guest/user_work/build/tmp/work-shared/aprzg0a/kernel-source/drivers/net/wirele
ss/wmp10/wmp10_apis/apis/ble/core/include -DRSI_HAL -Wl,-O1 -Wl,--hash-style=gnu -W
l,--as-needed wmp10_ble_sample.c -o wmp10_ble_sample
```

ビルドが成功しますと、『wmp10\_ble\_sample』が作成されます。

## 5. 動作確認

本章では、無線 LAN モジュールを動作させる手順を説明します。

### 5.1 Linux の起動

無線 LAN モジュールドライバを追加した Linux の起動方法を説明します。なお、Linux カーネルを microSD カードに書き込む方法は、AP-RZG-0A の Linux 開発キット (LK-RZG-A02) のソフトウェアマニュアルでご確認ください。

- ① AP-RZG-0A ボードの CN4 に WM-RP-10 を接続します。  
接続方法に関しては、無線 LAN モジュールの『ハードウェアマニュアル』でご確認ください。
- ② 『[2. Linux カーネルへの対応方法](#)』の Linux カーネルが書き込まれた microSD カードをスロットに挿入した後に、電源を入れます。

### 5.2 WLAN

#### 5.2.1 インフラストラクチャモード

無線 LAN モジュールをインフラストラクチャモードで使用する手順を説明します。  
なお、接続するアクセスポイントは以下の表の設定とします。

設定項目	設定値
SSID	WLAN-SSID
パスワード	wlan-pass
セキュリティモード	WPA2
チャンネル	1

Table5.2.1 アクセスポイントの設定例

- ① root でログインします。

```
aprzg0a login: root
```

- ② iwconfig の設定項目を設定します。

```
# iwconfig wlan0 mode Managed channel 0 essid WLAN-SSID key s:wlan-pass
```

- ③ セキュリティモードを「2(WPA2)」に設定します。

```
# iwpriv wlan0 setsecuritymode 2
```

- ④ Region コードを「3(JP)」に設定します。

```
# iwpriv wlan0 setregion 3
```

iwpriv の設定項目については、『Table3.1.2 iwpriv で設定可能なパラメータ』を参考に値を設定してください。

- ⑤ IP アドレス等を設定します。

```
# ifconfig wlan0 192.168.128.203 netmask 255.255.255.0 ←入力
```

上記の設定を行うことでアクセスポイントに接続できます。

接続後は、wlan0 に対して通信をすることで動作できます。例としては、ping を用いた方法を説明します。

- ⑥ アクセスポイントに接続されている機器(IP アドレスは、192.168.128.210 と仮定)に対して ping 送信で通信確認をします。

```
# ping -I wlan0 192.168.128.210 ←入力
```

## 5.2.2 アクセスポイントモード

無線 LAN モジュールをアクセスポイントモードで使用する手順を説明します。

なお、生成するアクセスポイントは以下の表の設定とします。

設定項目	設定値
SSID	WLAN-SSID
パスワード	wlan-pass
セキュリティモード	WPA2
チャンネル	1
リージョン	JP
Hidden AP	無効
アンテナ設定	オンボードアンテナ

Table5.2.2 WM-RP-10 の設定

- ① root でログインします。

```
aprzg0a login: root ←入力
```

- ② iwconfig の設定項目を設定します。

```
# iwconfig wlan0 mode Master channel 1 essid WLAN-SSID key s:wlan-pass ←入力
```

- ③ セキュリティモードを「2(WPA2)」に設定します。

```
# iwpriv wlan0 setsecuritymode 2 ←入力
```

- ④ リージョンコードを「3(JP)」に設定します。

```
# iwpriv wlan0 setregion 3 ←入力
```

- ⑤ Hidden AP 設定を「0(無効)」に設定します。

```
# iwpriv wlan0 sethidden 0 ←入力
```

- ⑥ アンテナ設定を「0(オンボードアンテナ)」に設定します。

```
# iwpriv wlan0 setantenna 0 ←入力
```

- ⑦ IP アドレス等を設定します。

```
# ifconfig wlan0 192.168.128.203 netmask 255.255.255.0 ←入力
```

上記の設定を行うことでアクセスポイントを生成できます。



- ⑧ 機器から WMRP-10 に接続し、機器(IP アドレスは、192.168.128.210 と仮定)に対して ping 送信で通信確認をします。

```
# ping -I wlan0 192.168.128.210
```

### 5.2.3 WPS モード(Push button)

無線 LAN モジュールを WPS モードで使用する手順を説明します。  
 なお、接続するアクセスポイントは、以下の表の設定とします。

設定項目	設定値
SSID	WLAN-SSID
パスワード	wlan-pass
セキュリティモード	WPA2
チャンネル	1
WPS	有効(Push button)

Table5.2.3 アクセスポイントの設定例

- ① root でログインします。

```
aprzg0a login: root
```

- ② iwconfig の設定項目を設定します。

```
# iwconfig wlan0 mode Managed channel 0
```

- ③ リージョンコードを「3(JP)」に設定します。

```
# iwpriv wlan0 setregion 3
```

- ④ WPS 設定を「1(Push button)」に設定します。

```
# iwpriv wlan0 setwpsmode 1
```

- ⑤ IP アドレス等を設定します。

```
# ifconfig wlan0 192.168.128.203 netmask 255.255.255.0
```

※ ②で指定したモードに完了後の動作が異なります。

「Managed」：アクセスポイントへの接続は開始されません。

「Master」：通常のアクセスポイントが生成されます。WPS 接続についてはこの時点では開始されません。

- ⑥ WPS 接続を開始します。

```
# iwpriv wlan0 wps_start
```

※ 接続に失敗した場合はコンソール上で『ifconfig wlan0 down』と入力し、手順⑤からやり直してください。

- ⑦ アクセスポイント側の WPS (Push button 方式)の接続を開始します。

- ⑧ アクセスポイントに接続されている機器(IP アドレスは、192.168.128.210 と仮定)に対して ping 送信で通信確認をします。

```
# ping -I wlan0 192.168.128.210
```

## 5.2.4 WPS モード (PIN 入力)

無線 LAN モジュールを WPS(インフラストラクチャ)で使用する手順を説明します。  
 なお、接続するアクセスポイントは、以下の表の設定とします。

設定項目	設定値
SSID	WLAN-SSID
パスコード	wlan-pass
セキュリティモード	WPA2
チャンネル	1
WPS	有効(PIN 入力)

Table5.2.4 アクセスポイントの設定例

① root でログインします。

```
aprzg0a login: root
```

② iwconfig の設定項目を設定します。

```
# iwconfig wlan0 mode Managed channel 0
```

③ リージョンコードを「3(JP)」に設定します。

```
# iwpriv wlan0 setregion 3
```

④ WPS 設定を「2(PIN 手動入力)」に設定します。

```
# iwpriv wlan0 setwpsmode 2
```

⑤ 使用する PIN コードを設定します。

```
# iwpriv wlan0 setwpspin xxxxxxxx
```

⑥ IP アドレス等を設定します。

```
# ifconfig wlan0 192.168.128.203 netmask 255.255.255.0
```

⑦ WPS 接続を開始します。

```
# iwpriv wlan0 wps_start
```

※ 接続に失敗した場合はコンソール上で『ifconfig wlan0 down』と入力し、手順⑤からやり直してください。

⑧ ⑤で指定した PIN コードを使用し、アクセスポイント側の WPS (PIN 方式)の接続を開始します。

⑨ アクセスポイントに接続されている機器(IP アドレスは、192.168.128.210 と仮定)に対して ping 送信で通信確認をします。

```
# ping -I wlan0 192.168.128.210
```

## 5.2.5 WPS モード (PIN 自動生成)

無線 LAN モジュールを WPS(インフラストラクチャ)で使用する手順を説明します。  
 なお、接続するアクセスポイントは、以下の表の設定とします。

設定項目	設定値
SSID	WLAN-SSID
パスコード	wlan-pass
セキュリティモード	WPA2
チャンネル	1
WPS	有効(PIN 入力)

Table5.2.5 アクセスポイントの設定例

① root でログインします。

```
aprzg0a login: root
```

② iwconfig の設定項目を設定します。

```
# iwconfig wlan0 mode Managed channel 0
```

③ リージョンコードを「3(JP)」に設定します。

```
# iwpriv wlan0 setregion 3
```

④ WPS 設定を「3(PIN 自動生成)」に設定します。

```
# iwpriv wlan0 setwpsmode 3
```

⑤ IP アドレス等を設定します。

```
# ifconfig wlan0 192.168.128.203 netmask 255.255.255.0
```

成功すると WM-RP-10 が設定した PIN コードが表示されます。

```
Generate PIN:xxxxxxx
```

⑥ WPS 接続を開始します。

```
# iwpriv wlan0 wps_start
```

※ 接続に失敗した場合はコンソール上で『ifconfig wlan0 down』と入力し、手順⑤からやり直してください。

⑧ ⑤で取得した PIN コードを使用し、アクセスポイント側の WPS (PIN 方式)の接続を開始します。

⑨ アクセスポイントに接続されている機器(IP アドレスは、192.168.128.210 と仮定)に対して ping 送信で通信確認をします。

```
# ping -I wlan0 192.168.128.210
```

## 5.2.6 EAP モード

無線 LAN モジュールを EAP(インフラストラクチャ)で使用する手順を説明します。  
 なお、接続するアクセスポイントは、以下の表の設定とします。

設定項目	設定値
SSID	WLAN-SSID
パスワード	wlan-pass
セキュリティモード	WPA2
チャンネル	1
EAP	有効(PEAP)
認証方式	MSCHAPV2
EAP ユーザ名	test-user
EAP パスワード	eap-pass

Table5.2.6 アクセスポイントの設定例

- ① root でログインします。

```
aprzg0a login: root
```

- ② iwconfig の設定項目を設定します。

```
# iwconfig wlan0 mode Managed channel 0 essid WLAN-SSID key s:wlan-pass
```

- ③ セキュリティモードを「5(EAP-WPA2)」に設定します。

```
# iwpriv wlan0 setsecuritymode 5
```

- ④ リージョンコードを「3(JP)」に設定します。

```
# iwpriv wlan0 setregion 3
```

- ⑤ EAP の動作設定を「PEAP」に設定します。

```
# iwpriv wlan0 seteapmethod PEAP
```

- ⑥ EAP の認証方式を「MSCHAPV2」に設定します。

```
# iwpriv wlan0 seteapinner "$"auth=MSCHAPV2$"
```

- ⑦ ユーザ名を「test-user」に設定します。

```
# iwpriv wlan0 setusername "$"test-user$"
```

- ⑧ パスワードを「eap-pass」に設定します。

```
# iwpriv wlan0 setpassword "$"eap-pass$"
```

- ⑨ IP アドレス等を設定します。

```
# ifconfig wlan0 192.168.128.203 netmask 255.255.255.0
```

- ⑩ アクセスポイントに接続されている機器(IP アドレスは、192.168.128.210 と仮定)に対して ping 送信で通信確認をします。

```
# ping -I wlan0 192.168.128.210
```

## 5.3 BT

### 5.3.1 Master モード

サンプルプログラムを使用し、無線 LAN モジュールを BT Master モードで使用する手順を説明します。  
 なお、WM-RP-10 は、以下の表の設定とします。

設定項目	設定値
デバイス名	BT-MASTER
パスワード	1234
接続先のデバイス名	BT-SLAVE

Table5.3.1 WM-RP-10 の設定例

- ① root でログインします。

```
aprzg0a login: root ←入力
```

- ② 『4. Bluetooth サンプルアプリのビルド』で作成した wmrp10\_bt\_sample をターゲットボードのファイルシステムにコピーします。

- ③ BT 動作確認用のサンプルアプリケーションを動作させます。

【Table5.3.1 WM-RP-10 の設定例】の設定内容をオプションで指定し、slave デバイスへの接続を行います。

```
# ./wmp10_bt_sample -m 1 -o BT-MASTER -l BT-SLAVE -p 1234 ←入力
```

- ④ SPP データの送信を行います。

```
# ./wmp10_bt_sample -s 12345678 ←入力
```

指定した SPP データを接続先のデバイスで受信できることを確認してください。

- ⑤ SPP データの受信を行います。

コマンド実行からパラメータで指定した時間が経過するまで、受信した SPP データを画面に表示します。  
 下記のコマンドでは実行から 10 秒間データ受信が有効になります。

```
# ./wmp10_bt_sample -r 10 ←入力
```

```
x Bytes receive : xxxxxxxx
```

- ⑥ Slave デバイスとの接続を停止します。「xx:xx:xx:xx:xx:xx」には接続中の Slave デバイスの BD アドレスを入力してください。

```
# ./wmp10_bt_sample -d xx:xx:xx:xx:xx:xx ←入力
```

## 5.3.2 Slave モード

サンプルプログラムを使用し、無線 LAN モジュールを BT Slave モードで使用する手順を説明します。  
 なお、WM-RP-10 は、以下の表の設定とします。

設定項目	設定値
デバイス名	BT-SLAVE
パスコード	1234

Table5.3.2 WM-RP-10 の設定例

- ① root でログインします。

```
aprzg0a login: root
```

- ② 『4. Bluetooth サンプルアプリのビルド』で作成した wmrp10\_bt\_sample をターゲットボードのファイルシステムにコピーします。

- ③ BT 動作確認用のサンプルアプリケーションを動作させます。

【Table5.3.2 WM-RP-10 の設定例】の設定内容をオプションで指定し、slave デバイスへの接続を行います。

```
# ./wmp10_bt_sample -m 2 -o BT-SLAVE -p 1234
```

- ④ Master デバイスからモジュールに接続します。

- ⑤ SPP データの送信を行います。

```
# ./wmp10_bt_sample -s 12345678
```

指定した SPP データを接続先のデバイスで受信できることを確認してください。

- ⑥ SPP データの受信を行います。

コマンド実行からパラメータで指定した時間が経過するまで、受信した SPP データを画面に表示します。

下記のコマンドでは実行から 10 秒間表示が有効になります。

```
# ./wmp10_bt_sample -r 10
```

```
x Bytes receive : xxxxxxxx
```

- ⑦ Master デバイスとの接続を停止します。「xx:xx:xx:xx:xx:xx」には接続中の Master デバイスの BD アドレスを入力してください。

```
# ./wmp10_bt_sample -d xx:xx:xx:xx:xx:xx
```

## 5.4 BTLE

### 5.4.1 BTLE Central モード

サンプルプログラムを使用し、無線 LAN モジュールを BLE Central モードで使用する手順を説明します。  
 なお、WM-RP-10 は、以下の表の設定とします。

設定項目	設定値
デバイス名	BLE-CENTRAL
接続先のデバイス名	BLE- PERIPHERAL
BD アドレス	11:22:33:44:55:66

Table5.4.1 WM-RP-10 の設定例

- ① root でログインします。

```
aprzg0a login: root
```

- ② 『4. Bluetooth サンプルアプリのビルド』で作成した wmrp10\_ble\_sample をターゲットボードのファイルシステムにコピーします。

- ③ BT 動作確認用のサンプルアプリケーションを動作させます。

【Table5.4.1 WM-RP-10 の設定例】の設定内容をオプションで指定し、Peripheral デバイスへの接続を行います。

```
# ./wmp10_ble_sample -m 1 -o BLE-CENTRAL -l BLE-PERIPHERAL -d 11:22:33:44:55:66
```

- ④ WM-RP-10 が Peripheral デバイスへ接続します。

接続に成功するとコンソールに以下のような画面が表示されます。

```
BLE CENTRAL:
```

- ⑤ 接続したデバイスのバッテリー情報を取得します。

```
BLE CENTRAL: g
```

- ⑥ バッテリー情報の取得が成功すると以下のような画面が表示されます。

```
get_data = xxx
```

- ⑦ Peripheral デバイスとの接続を停止します。

Peripheral デバイス側で切断されていることを確認してください。

```
BLE CENTRAL: e
```

## 5.4.2 BLE Peripheral モード

サンプルプログラムを使用し、無線 LAN モジュールを BLE Peripheral モードで使用する手順を説明します。  
 なお、WM-RP-10 は、以下の表の設定とします。

設定項目	設定値
デバイス名	BLE-PERIPHERAL
接続先のデバイス名	BLE-CENTRAL
BD アドレス	11:22:33:44:55:66

Table5.4.2 WM-RP-10 の設定例

- ① root でログインします。

```
aprzg0a login: root ↵
```

- ② 『4. Bluetooth サンプルアプリのビルド』で作成した wmrp10\_ble\_sample をターゲットボードのファイルシステムにコピーします。

- ③ BT 動作確認用のサンプルアプリケーションを動作させます。

【Table5.4.2 WM-RP-10 の設定例】の設定内容をオプションで指定し、アドバタイズを開始します。

```
# ./wmp10_ble_sample -m 2 -o BLE-PERIPHERAL -d 11:22:33:44:55:66 ↵
```

- ④ アドバタイズに成功するとコンソールに以下のような画面が表示されます。

```
BLE PERIPHERAL :
```

- ⑤ WM-RP-10 のバッテリー情報を設定します。

例として「50」と入力した場合、WM-RR-10 のバッテリー情報が 50%として設定されます。

```
BLE PERIPHERAL :50 ↵
```

- ⑥ Central デバイスから Scan を行い、「BLE-PERIPHERAL」というデバイスに接続してください。  
 接続後、バッテリー情報が④で設定したとおりに変更されていることを確認してください。

- ⑦ WM-RP-10 のアドバタイズ動作を停止します。

```
BLE PERIPHERAL :e ↵
```

- ⑧ Central デバイスから Scan できないことを確認してください。



## ご注意

- ・本文書の著作権は、株式会社アルファプロジェクトが保有します。
- ・本文書の内容を無断で転載することは一切禁止します。
- ・本文書に記載されているサンプルプログラムの著作権は、株式会社アルファプロジェクトが保有します。
- ・本文書に記載されている内容およびサンプルプログラムについての技術サポートは一切受け付けておりません。
- ・本文書の内容およびサンプルプログラムに基づき、アプリケーションを運用した結果、万一損害が発生しても、弊社では一切責任を負いませんのでご了承下さい。
- ・本文書の内容については、万全を期して作成いたしましたが、万一ご不審な点、誤りなどお気付きの点がありましたら弊社までご連絡下さい。
- ・本文書の内容は、将来予告なしに変更されることがあります。

## 商標について

- ・R8A7745 は、ルネサスエレクトロニクス株式会社の登録商標、商標または商品名称です。
- ・Linux は、Linus Torvalds の米国およびその他の国における登録商標または商標です。
- ・Yocto Project は、Linux Foundation の登録商標です。
- ・U-Boot は、DENX Software Engineering の登録商標、商標または商品名称です。
- ・VirtualBox は、Oracle Corporation の商品名称です。
- ・Windows®の正式名称は、Microsoft®Windows®Operating System です。
- ・Microsoft、Windows は、米国 Microsoft Corporation.の米国およびその他の国における商標または登録商標です。
- ・Windows®10、Windows®7 は、米国 Microsoft Corporation.の商品名称です。  
本文書では下記のように省略して記載している場合がございます。ご了承下さい。  
Windows®10 は、Windows 10 もしくは Win10  
Windows®7 は、Windows 7 もしくは Win7
- ・その他の会社名、製品名は、各社の登録商標または商標です。