

MS104-SH4AG

USB 無線 LAN モジュールの使用方法

1版 2009年07月20日

1. 概要

本アプリケーションノートは、USB 無線 LAN モジュールを MS104-SH4AG で使用方法について説明します。

本アプリケーションノートは、「MS104-SH4AG Linux 開発キット Linux-Kit-C01 Release 2.1」がインストールされていることを前提とします。

Linux のバージョンは linux-2.6.29.4-alp-2.1 以上である必要があります。

対応している USB 無線 LAN モジュールは以下のとおりとなります。

- ・GW-US54GXS (Planex 社)

2. システムの変更

本章では USB 無線 LAN モジュールを使用するために必要となるシステムの変更について解説します。例として、RAMFS-Linux システムをベースとして進めます。

2.1 ルートファイルシステムの変更

無線 LAN ユーティリティ及びファームウェアをインストールするために Buildroot を編集し、ファイルシステムを再作成します。

Buildroot は自動的にインターネット上からファイルをダウンロードするため、ゲスト OS をインターネットへアクセス可能な状態にしておく必要があります。

- ① ゲスト OS を起動し、Buildroot のディレクトリに移動します。

```
[guest@LinuxKit ~]$ cd linuxkit-ms104sh4ag/buildroot-2009.05-alp
[guest@LinuxKit buildroot-2009.05-alp]$
```

- ② Buildroot のコンフィギュレータを起動します。

```
[guest@LinuxKit ~]$ make menuconfig
```

- ③ 『Package Selection for the target --->』へ進みます。

```
+-----+
| Target Architecture (superh) ---> |
| Target Architecture Variant (sh4) ---> |
| Target options ---> |
| Build options ---> |
| Toolchain ---> |
| Package Selection for the target ---> |
| Target filesystem options ---> |
| Kernel ---> |
| --- |
| Load an Alternate Configuration File |
| Save an Alternate Configuration File |
+-----+
```

- ④ 『Networking --->』へ進みます。

```
+^(-)-----+
| [ ] shared-mime-info
| [ ] startup-notification
| [ ] strace
| [ ] sudo
| Database --->
| Networking --->
| Hardware handling / blockdevices and filesystem maintenance
| Audio and video libraries and applications --->
| Graphic libraries and applications (graphic/text) --->
| Compressors / decompressors --->
| Package managers --->
+---v(+)-----+
```

- ⑤ 『wireless tools』にチェックを入れます。

```
+^(-)-----+
| [ ] dhcpdump
| [ ] tftpd
| [ ] tn5250
| [ ] ttcp
| [ ] udpcast
| [ ] vpnc
| [*] vsftpd
| [ ] vtun - BEWARE: read package/vtun/README.txt before use
| [ ] webif - Status Console
| [*] wireless tools
| [ ] wpa_supplicant
+-----+
```

- ⑥ Buildroot のコンフィギュレータを終了し、Busybox のコンフィギュレータを起動します。

```
[guest@LinuxKit buildroot-2009.05-alp]$ make busybox-menuconfig
```

- ⑦ 『Linux System Utilities --->』へ進みます。

```
+^(-)-----+
| Coreutils --->
| Console Utilities --->
| Debian Utilities --->
| Editors --->
| Finding Utilities --->
| Init Utilities --->
| Login/Password Management Utilities --->
| Linux Ext2 FS Progs --->
| Linux Module Utilities --->
| Linux System Utilities --->
| Miscellaneous Utilities --->
+v(+)-----+
```

- ⑧ 『Support loading of firmwares』にチェックを入れ、コンフィギュレータを終了します。

```
+^(-)-----+
| [*] mdev
| [*] Support /etc/mdev.conf
| [*] Support subdirs/symlinks
| [*] Support regular expressions substitutions when renaming dev
| [*] Support command execution at device addition/removal
| [*] Support loading of firmwares
| [*] mkswap
| [ ] Version 0 support
| [*] more
| [*] Use termios to manipulate the screen
| [*] mount
+v(+)-----+
```

- ⑨ ルートファイルシステムにファームウェアをインストールします。ファームウェアを配置するためのディレクトリを作成します。

```
[guest@LinuxKit buildroot-2009.05-alp]$ mkdir -p project_build_sh4/ms104sh4ag/root/lib/firmware
[guest@LinuxKit buildroot-2009.05-alp]$
```

- ⑩ ゲストOSのファームウェアをルートファイルシステムにコピーします。

```
[guest@LinuxKit buildroot-2009.05-alp]$ cp -a /lib/firmware/zd1211/ project_build_sh4/ms104sh4ag/root/lib/firmware/
[guest@LinuxKit buildroot-2009.05-alp]$
```

- ⑪ make を実行します。

```
[guest@LinuxKit buildroot-2009.05-alp]$ make
...
```

- ⑫ 作成された『rootfs.sh4-ms104sh4ag.cpio.gz』を/tftpbootディレクトリにコピーします。

```
[guest@LinuxKit buildroot-2009.05-alp]$ cp binaries/ms104sh4ag/rootfs.sh4-ms104sh4ag.cpio.gz /tftpboot
...
```

2.2 Linux カーネルの変更

- ① Linux カーネルのディレクトリに移動します。

```
[guest@LinuxKit ~]$ cd linuxkit-ms104sh4ag/linux-2.6.29.4-alp
[guest@LinuxKit linux-2.6.29.4-alp]$
```

- ② RAMFS-Linux システムの設定を呼び出します。

```
[guest@LinuxKit linux-2.6.29.4-alp]$ ARCH=sh make ms104sh4ag_ramfs_defconfig
#
# configuration written to .config
#
[guest@LinuxKit linux-2.6.29.4-alp]$
```

- ③ コンフィギュレータを起動します。

```
[guest@LinuxKit linux-2.6.29.4-alp]$ ARCH=sh make menuconfig
```

```
.config - buildroot v2009.05 Configuration
-----
+----- Buildroot Configuration -----+
| Arrow keys navigate the menu. <Enter> selects submenus --->. |
| Highlighted letters are hotkeys. Pressing <Y> selectes a feature, |
| while <N> will exclude a feature. Press <Esc><Esc> to exit, <?> for |
| Help, </> for Search. Legend: [*] feature is selected [ ] feature is |
+-----+
| Target Architecture (superh) ---> |
| Target Architecture Variant (sh4) ---> |
| Target options ---> |
| Build options ---> |
| Toolchain ---> |
| Package Selection for the target ---> |
| Target filesystem options ---> |
| Kernel ---> |
| --- |
| Load an Alternate Configuration File |
| Save an Alternate Configuration File |
+-----+
| <Select> < Exit > < Help > |
+-----+
```

- ④ 『Networking support --->』へ進みます。

```
+-----+
|                                     |
|   General setup --->               |
|   [*] Enable loadable module support ---> |
|   [*] Enable the block layer --->      |
|   System type --->                 |
|   Kernel features --->              |
|   Boot options --->                |
|   Bus options --->                 |
|   Executable file formats --->       |
|   Power management options (EXPERIMENTAL) ---> |
|   [*] Networking support --->      |
|   Device Drivers --->              |
+-----v(+)-----+
```

- ⑤ 『Wireless --->』にチェックを入れ進みます。

```
+-----+
|   --- Networking support           |
|   Networking options --->         |
|   [ ] Amateur Radio support --->   |
|   < > CAN bus subsystem support ---> |
|   < > IrDA (infrared) subsystem support ---> |
|   < > Bluetooth subsystem support ---> |
|   < > RxRPC session sockets         |
|   < > Phonet protocols family       |
|   [*] Wireless --->                |
|   < > WiMAX Wireless Broadband support ---> |
|   < > RF switch subsystem support ---> |
+-----v(+)-----+
```

- ⑥ 『Generic IEEE 802.11 Networking Stack (mac80211)』を選択します。自動的に他の項目にもチェックがはまります。

```
+-----+
|   --- Wireless                     |
|   -* Improved wireless configuration API |
|   [ ] cfg80211 regulatory debugging     |
|   [*] nl80211 new netlink interface support |
|   [*] Old wireless static regulatory definitions |
|   -* Wireless extensions              |
|   [*] Wireless extensions sysfs files   |
|   < > Common routines for IEEE802.11 drivers |
|   <*> Generic IEEE 802.11 Networking Stack (mac80211) |
|   Rate control algorithm selection ---> |
|   [ ] Enable mac80211 mesh networking (pre-802.11s) support |
+-----v(+)-----+
```

- ⑦ トップの階層に戻り 『Device Drivers --->』へ進みます。

```
+-----^(-)-----+
|   [*] Enable loadable module support ---> |
|   [*] Enable the block layer --->      |
|   System type --->                 |
|   Kernel features --->              |
|   Boot options --->                |
|   Bus options --->                 |
|   Executable file formats --->       |
|   Power management options (EXPERIMENTAL) ---> |
|   [*] Networking support --->      |
|   Device Drivers --->            |
|   File systems --->                |
+-----v(+)-----+
```

- ⑧ 『Network device support --->』へ進みます。

```
+-----^(-)-----+
| < > Connector - unified userspace <-> kernelspace linker ---> |
| <*> Memory Technology Device (MTD) support ---> |
| < > Parallel port support ---> |
| [*] Block devices ---> |
| [ ] Misc devices ---> |
| <*> ATA/ATAPI/MFM/RLL support ---> |
|     SCSI device support ---> |
| < > Serial ATA (prod) and Parallel ATA (experimental) drivers -- |
| [ ] Multiple devices driver support (RAID and LVM) ---> |
| [*] Network device support ---> |
| [ ] ISDN support ---> |
+-----v(+)-----+
```

- ⑨ 『Wireless LAN --->』へ進みます。

```
+-----^(-)-----+
| < > Bonding driver support |
| < > MAC-VLAN support (EXPERIMENTAL) |
| < > EQL (serial line load balancing) support |
| < > Universal TUN/TAP device driver support |
| < > Virtual ethernet pair device |
| -* PHY Device support and infrastructure ---> |
| [*] Ethernet (10 or 100Mbit) ---> |
| [ ] Ethernet (1000 Mbit) ---> |
| [ ] Ethernet (10000 Mbit) ---> |
| Wireless LAN ---> |
| *** Enable WiMAX (Networking options) to see the WiMAX driv |
+-----v(+)-----+
```

- ⑩ 『Wireless LAN (IEEE 802.11)』にチェックを入れ、『ZyDAS ZD1211/ZD1211B USB-wireless support』にもチェックを入れます。

```
+-----^(-)-----+
| [ ] Wireless LAN (pre-802.11) |
| [*] Wireless LAN (IEEE 802.11) |
| < > Marvell 8xxx Libertas WLAN driver support |
| < > Marvell 8xxx Libertas WLAN driver support with thin firmwar |
| < > USB ZD1201 based Wireless device support |
| < > Wireless RNDIS USB support |
| < > Realtek 8187 and 8187B USB support |
| < > Simulated radio testing tool for mac80211 |
| < > Softmac Prism54 support |
| < > IEEE 802.11 for Host AP (Prism2/2.5/3 and WEP/TKIP/CCMP) |
| < > Broadcom 43xx wireless support (mac80211 stack) |
| < > Broadcom 43xx-legacy wireless support (mac80211 stack) |
| <*> ZyDAS ZD1211/ZD1211B USB-wireless support |
| [ ] ZyDAS ZD1211 debugging |
+-----v(+)-----+
```

- ⑪ コンフィギュレータを終了し、テキストエディタでファイルを編集します。

net/wireless/reg.c 114 行目付近の「US」を「JP」に変更してください。

変更前

```
#ifdef CONFIG_WIRELESS_OLD_REGULATORY
static char *ieee80211_regdom = "US";
module_param(ieee80211_regdom, charp, 0444);
MODULE_PARM_DESC(ieee80211_regdom, "IEEE 802.11 regulatory domain code");
```

変更後

```
#ifdef CONFIG_WIRELESS_OLD_REGULATORY
static char *ieee80211_regdom = "JP";
module_param(ieee80211_regdom, charp, 0444);
MODULE_PARM_DESC(ieee80211_regdom, "IEEE 802.11 regulatory domain code");
```

drivers/net/wireless/zd1211rw/zd_mac.c 45 行目付近に 「{ 0x49, "JP" },」 を挿入してください。

変更前

```
{ ZD_REGDOMAIN_JAPAN, "JP" },  
{ ZD_REGDOMAIN_JAPAN_ADD, "JP" },  
{ ZD_REGDOMAIN_SPAIN, "ES" },  
{ ZD_REGDOMAIN_FRANCE, "FR" },
```

変更後

```
{ ZD_REGDOMAIN_JAPAN, "JP" },  
{ ZD_REGDOMAIN_JAPAN_ADD, "JP" },  
{ 0x49, "JP" },  
{ ZD_REGDOMAIN_SPAIN, "ES" },  
{ ZD_REGDOMAIN_FRANCE, "FR" },
```

- ⑫ make を実行します。

```
[guest@LinuxKit linux-2.6.29.4-alp]$ ARCH=sh CROSS_COMPILE=sh4-linux- make uImage  
...
```

- ⑬ make が正常に終了すると、uImage が作成されます。

```
[guest@LinuxKit linux-2.6.29.4-alp]$ ls arch/sh/boot/uImage  
arch/sh/boot/uImage  
[guest@LinuxKit linux-2.6.29.4-alp]$
```

3. 無線 LAN の設定

前章で作成された Linux 上でアクセスポイントに接続し、通信する手順を説明します。ESSID やチャンネルなどの無線 LAN ネットワークの設定には『iwconfig』コマンドを、IP アドレスやサブネットマスクの設定には『ifconfig』コマンドを使用します。

項目名	値
ESSID	TEST-AP
暗号化モード	WEP
WEP キー	00112233445566778899AABBCC
チャンネル	13
アクセスポイント	00:11:22:33:44:55
IP アドレス	192.168.128.203
サブネットマスク	255.255.255.0

Table3.1-1 ネットワーク設定値

実際に運用する際の設定値は、ネットワーク管理者にお問い合わせ下さい。

- ① 『Linux 開発キットソフトウェアマニュアル U-boot 編』を参照し、MS104-SH4AG とお使いの PC を接続し、MS104-SH4AG の電源を投入します。前章で作成した Linux を起動しログインしてください。

- ② iwconfig コマンドを実行し、wlan0 インタフェースが存在することを確認します。

```
# iwconfig
lo          no wireless extensions.

eth0       no wireless extensions.

wmaster0   no wireless extensions.

wlan0      IEEE 802.11bg  ESSID:""
           Mode:Managed  Frequency:2.412 GHz  Access Point: Not-Associated
           Tx-Power=0 dBm
           Retry min limit:7   RTS thr:off   Fragment thr=2352 B
           Encryption key:off
           Power Management:off
           Link Quality:0  Signal level:0  Noise level:0
           Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
           Tx excessive retries:0  Invalid misc:0  Missed beacon:0

#
```

- ③ iwconfig コマンドで wlan0 インタフェースの ESSID、チャンネル及び WEP キーを設定します。

```
# iwconfig wlan0 essid TEST-AP channel 13 key 00112233445566778899AABBCC
#
```

- ④ ifconfig コマンドで wlan0 インタフェースに IP アドレスとサブネットマスクを設定します。

```
# ifconfig wlan0 192.168.128.203 netmask 255.255.255.0
usb 1-1: firmware: requesting zd1211/zd1211b_ub
usb 1-1: firmware: requesting zd1211/zd1211b_uphr
zd1211rw 1-1:1.0: firmware version 4725
zd1211rw 1-1:1.0: zd1211b chip 2019:5303 v4810 high 00-22-cf AL2230_RF pa0 ---N-
cfg80211: Calling CRDA for country: JP
#
```

- ⑤ 再度 iwconfig コマンドを実行して wlan0 インタフェースにアクセスポイントが設定されていることを確認します。

```
# iwconfig
lo          no wireless extensions.

eth0       no wireless extensions.

wmaster0   no wireless extensions.

wlan0      IEEE 802.11bg  ESSID:"TEST-AP"
           Mode:Managed  Frequency:2.472 GHz  Access Point: 00:11:22:33:44:55
           Bit Rate=1 Mb/s   Tx-Power=20 dBm
           Retry min limit:7   RTS thr:off   Fragment thr=2352 B
           Encryption key:0011-2233-4455-6677-8899-AABB-CC   Security mode:open
           Power Management:off
           Link Quality=100/100  Signal level:52/100
           Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
           Tx excessive retries:0  Invalid misc:0  Missed beacon:0

#
```

4. サンプルプログラム

ネットワークインタフェースを指定して通信をする簡単なサンプルプログラムを作成します。

指定した IP アドレスのポート番号に接続し、受信したデータを表示します。今回はゲスト OS 上で DAYTIME サーバを起動し、ネットワークインタフェース wlan0 を経由して、情報を取得します。

プログラムに指定する引数は以下のようになります。

- ・第1引数：ネットワークインタフェース名
- ・第2引数：IP アドレス又はホスト名
- ・第3引数：ポート番号又はサービス名

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netdb.h>
#include <net/if.h>

static int
setsockopt_bindtodevice(int fd, const char *iface)
{
    int r;
    struct ifreq ifr;

    strncpy(ifr.ifr_name, iface, IFNAMSIZ);
    if ((r = setsockopt(fd, SOL_SOCKET, SO_BINDTODEVICE, &ifr, sizeof(ifr)))
        fprintf(stderr, "can't bind to interface %s\n", iface);
    return r;
}

int
main(int argc, char **argv)
{
    int fd, n, s;
    struct addrinfo hints;
    struct addrinfo *result, *rp;
    const char *iface = argv[1];
    const char *host = argv[2];
    const char *port = argv[3];
    char buf[BUFSIZ];

    if (argc != 4) {
        fprintf(stderr, "Usage wireless-test <iface> <host> <port>\n");
    }
}
```



```

    exit(1);
}

memset(&hints, 0, sizeof(hints));
hints.ai_family = AF_UNSPEC;
hints.ai_socktype = SOCK_STREAM;
hints.ai_flags = 0;
hints.ai_protocol = 0;
if ((s = getaddrinfo(host, port, &hints, &result))) {
    fprintf(stderr, "getaddrinfo: %s¥n", gai_strerror(s));
    exit(1);
}

for (rp = result; rp ; rp = rp->ai_next) {
    fd = socket(rp->ai_family, rp->ai_socktype, rp->ai_protocol);

    setsockopt_bindtodevice(fd, iface);

    if (connect(fd, rp->ai_addr, rp->ai_addrlen) != -1)
        break;
    close(fd);
}

if (rp == NULL) {
    fprintf(stderr, "Could not connect¥n");
    exit(1);
}

freeaddrinfo(result);

while ((n = read(fd, buf, sizeof(buf)-1)) > 0) {
    buf[n] = '\0';
    printf("%s", buf);
}
return 0;
}

```

wireless-sample.c

4.1 ゲスト OS 上でのプログラムの作成

- ① ゲスト OS 上でサンプルプログラムのディレクトリを作成し、移動します。

```

[guest@LinuxKit ~]# mkdir linuxkit-ms104sh4ag/samples/wireless-sample
[guest@LinuxKit ~]# cd linuxkit-ms104sh4ag/samples/wireless-sample/
[guest@LinuxKit wireless-sample]#

```

- ② サンプルプログラム wireless-sample.c をコンパイルします。正常に終了すると実行ファイル wireless-sample が生成されます。

```

[guest@LinuxKit wireless-sample]# sh4-linux-gcc -o wireless-sample wireless-sample.c
[guest@LinuxKit wireless-sample]# ls wireless-sample
wireless-sample
[guest@LinuxKit wireless-sample]#

```

- ③ 作成した実行ファイルを/nfs ディレクトリにコピーします。

```

[guest@LinuxKit wireless-sample]# cp -a wireless-sample /nfs/
[guest@LinuxKit wireless-sample]#

```

4.2 ゲスト OS 上での DAYTIME サーバの起動

DAYTIME プロトコルは日付・時刻の情報を得るための通信プロトコルの 1 つで、TCP 又は UDP の 13 番ポートを使用します。結果は「10 JAN 2000 12:00:00 JST」のような形式で返されます。

本節はルート権限で作業する必要があります。

- ① xinetd の設定ファイル『/etc/xinetd.d/daytime-stream』を編集します。6 行目の「disable = yes」の行を「disable = no」に変更してください。

```
# This is the configuration for the tcp/stream daytime service.

service daytime
{
# This is for quick on or off of the service
  disable      = no

# The next attributes are mandatory for all services
  id           = daytime-stream
  type        = INTERNAL
  wait        = no
  socket_type = stream
# protocol    = socket type is usually enough
```

- ② 設定を反映させるため、xinetd をリロードします。

```
[root@LinuxKit ~]# /etc/init.d/xinetd reload
Reloading configuration: [ OK ]
[root@LinuxKit ~]#
```

4.3 サンプルプログラムの実行

- ① 『Linux 開発キットソフトウェアマニュアル U-boot 編』を参照し、MS104-SH4AG とお使いの PC を接続し、MS104-SH4AG の電源を投入します。

- ② 「第 3 章 無線 LAN の設定」を参照し、ネットワークインタフェース wlan0 を有効にします。

- ③ ログイン後、ゲスト OS の /nfs ディレクトリをマウントします。

```
# mount -t nfs -o nolock 192.168.128.201:/nfs /mnt/nfs
#
```

- ④ プログラムを実行します。

第 1 引数には無線 LAN ネットワークインタフェース wlan0、第 2 引数にはゲスト OS の IP アドレス 192.168.128.201、第 3 引数には DAYTIME プロトコルのポート番号 13 を指定します。

正常に終了すると、ゲスト OS の時刻が表示されます。

```
# /mnt/nfs/wireless-sample wlan0 192.168.128.201 13
22 JUL 2009 15:52:31 JST
#
```

ご注意

本文書の著作権は（株）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。

本文書の内容は、将来予告なしに変更されることがあります。

本文書に記載されている内容についての質問等のサポートは一切受け付けておりませんのでご了承ください。

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。万が一不審な点、誤りなどお気づきの点がありましたら弊社までご連絡下さい。

本文書の内容に基づき、アプリケーションを運用した結果、万が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。

商標について

- ・ SH7764 は、株式会社ルネサステクノロジの登録商標、商標または商品名称です。
- ・ Linux は、Linus Torvalds 氏の日本およびその他の国における登録商標または商標です。
- ・ VMware は米国およびその他の地域における VMware, Inc. の登録商標、商標または商品名称です。
- ・ 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® Vista、Windows® XP、Windows® 2000 Professional 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® Vista は Windows Vista もしくは WinVista
- ・ Windows® XP は Windows XP もしくは WinXP
- ・ Windows® 2000 Professional は Windows 2000 もしくは Win2000
- ・ 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。

 **ALPHA PROJECT Co.,LTD.**

株式会社アルファプロジェクト
〒431-3114
静岡県浜松市東区積志町 8 3 4
<http://www.apnet.co.jp>
E-MAIL : query@apnet.co.jp