

# ピセロ AK-130 ソフトウェアインストールガイド

株式会社エイビット

9 版 2025/4/28

## 1. はじめに

本書は、ワイヤレスマイクロサーバ「ピセロ AK-130」(以下、本機と記載) へのソフトウェアの導入について説明します。

### 1.1 用語

本書で使用する用語を説明します。

用語	説明
SoC	System on chip Raspberry Pi Compute Module 3 に搭載されているプロセッサ BCM2837 を指します。
SoM	System on module Raspberry Pi Comupte Module 3 を指します。
Raspberry Pi OS	Raspberry Pi 財団が配布する Raspberry Pi 向けの OS です。
差分パッケージ	弊社が本機デモ用に用意した Raspberry Pi OS に対する差分ファイル群です。

## 2. Raspberry Pi OS の利用

本機は Raspberry Pi Compute Module 3 Lite(以下、SoM と記載) を搭載しています。"Lite" は eMMC 未実装版のため、ソフトウェアは SD カードに格納します。本書では Raspberry Pi 向け OS "Raspberry Pi OS" をインストールする方法を示します。インストール手順は次の通りです。

1. Raspberry Pi OS イメージの入手と SD カードへの書き込み
2. Raspberry Pi OS の起動とシリアルコンソールからのログイン
3. 差分パッケージの書き込み
4. 動作確認

### 2.1 Raspberry Pi OS イメージの入手と SD カードへの書き込み

本書の手順で利用する Raspberry Pi OS のバージョンは次の通りです。

- Raspberry Pi OS Lite(64-bit)
- Release date: December 2024
- Kernel version: 6.6.64
- イメージファイル: 2024-11-19-raspios-bookworm-arm64-lite.img

後述する差分パッケージは、上記バージョンに対して有効です。

1. Raspberry Pi のウェブサイトより、OS 書き込みツール Raspberry Pi Imager をダウンロードしてインストールしてください。
2. Raspberry Pi Imager を起動し、ガイドに従って上記バージョンの OS を MicroSD カードに書き込んでください。このとき、「設定を編集する」を選択してユーザー名とパスワード等を適宜設定してください。
3. 起動パッケージを解凍し、SD カードのブートパーティション(bootfs)にコピーします。配布ファイル名は次の通りです。

ファイル名: ak-130-ax962\_boot\_20250121.zip

## 2.2 Raspberry Pi OS の起動とシリアルコンソールからのログイン

前節で作成した Raspberry Pi OS 入り MicroSD カードを本機の MicroSD カードスロットへ挿入し、電源を投入してください。

起動パッケージが正しく書き込まれていれば、Raspberry Pi OS 起動時に PWR LED が点滅します。

本機の MicroUSB-B 端子は、USB シリアル変換デバイスを介して、SoM の UART 端子に接続されており、Raspberry Pi OS のシリアルコンソールを利用することができます。本機の MicroUSB-B 端子と PC の USB 端子をケーブルで接続してください。

本機が搭載する USB シリアル変換デバイスはプロリフィック社の PL2303 です。PC においては PL2303 用のドライバをインストールしてください。Windows 向けドライバは、プロリフィック社のウェブサイトからダウンロードしてください。

PC においてシリアルターミナルソフトを起動してください。通信パラメータは次の通りです。

- ボーレート：115200bps
- データ長：8ビット
- ストップビット長：1ビット
- パリティ：なし
- フロー制御：なし

本機の電源を投入してからしばらくして、シリアルターミナルにログインプロンプトが表示されます。

```
Debian GNU/Linux 12 raspberrypi ttyAMA0
My IP address is XXX.XXX.XXX.XXX xxxx::xxxx:xxx:xxxx:xxxx
raspberrypi login:
```

前節の 2. で設定したユーザー名とパスワードでログインできます。

## 2.3 差分パッケージの書き込み

差分パッケージは、Raspberry Pi OS を本機で利用するための差分ファイル群です。本機向けに設定を変更した Linux カーネル、本機搭載デバイス用のデバインドライバおよび動作デモ用のスクリプトから成ります。PC 上で、差分パッケージファイルを弊社ウェブサイトからダウンロードし USB メモリに書き込んでください。配布ファイル名は次の通りです。

```
ファイル名: linux-6.6.64-v8-ak130+-arm64_20241226.tar.xz
```

USB メモリを本機 USB コネクタへ接続し、シリアルターミナル上から `dmesg` コマンドなどで USB メモリデバイスの検出を確認してください。

```
pi@raspberrypi:~$ dmesg
...
[ 1979.901977] sd 0:0:0:0: [sda] Assuming drive cache: write through
[ 1979.925129] sda: sda1
[ 1979.927623] sd 0:0:0:0: [sda] Attached SCSI removal disk
pi@raspberrypi:~$
```

USB メモリをファイルシステムにマウントし、差分パッケージをルートディレクトリへ展開してください。

```
pi@raspberrypi:~$ sudo mount /dev/sda1 /mnt
pi@raspberrypi:~$ cd /mnt
pi@raspberrypi:~$ sudo tar -C / --keep-directory-symlink -xJpvf /mnt/linux-6.6.64-v8-ak130+-arm64_20241226.tar.xz
pi@raspberrypi:~$ sync
```

## 2.4 動作確認

前節の差分パッケージ書き込みが完了した後、本機を再起動してください。

```
pi@raspberrypi:~$ sudo reboot
```

シリアルターミナルにシャットダウンおよび起動ログが表示され、再びログインプロンプトが表示されます。ログインして次のことを確認してください。

- Linux カーネルバージョン文字列

```
pi@raspberrypi:~$ uname -a
Linux raspberrypi 6.6.64-v8-ak130+ #10 SMP PREEMPT Fri Dec 20 14:10:30 CST 2024 aarch64
GNU/Linux
```

- 次章に示す搭載デバイスの機能確認

### 3. 搭載デバイスの利用

差分パッケージ適用後、本機搭載の各種デバイスが利用できるようになります。以下、デバイス毎の機能確認方法を示します。

#### 3.1 LED

本機前面に状態表示 LED が 3 種搭載されています。差分パッケージ適用後は LED クラスドライバにより sysfs 経由で制御できます。

ラベル	灯色	sysfs	説明
PWR	緑	/sys/class/leds/pwr_led	電源投入中は点灯がデフォルトです。 起動パッケージ適用後は、OS 稼働中は点滅します。
MON	黄	/sys/class/leds/mon_led	差分パッケージ適用後は、SDcard アクセス時に点灯します。
ALM	赤	/sys/class/leds/alm_led	差分パッケージ適用直後は未使用（消灯）

PWR の点滅を止めるには次のコマンドを実行してください。

```
pi@raspberrypi:~$ echo none | sudo tee /sys/class/leds/pwr_led/trigger
```

/boot/firmware/config.txt ファイルに起動時に PWR を点滅させるコマンドが記述されています。点滅が不要な場合は、以下の行の heartbeat を none に変更してください。

```
dtoverlay=gpio-led,gpio=13,label=pwr_led,trigger=heartbeat,active_low
```

/boot/firmware/config.txt ファイルに起動時に MON を点灯させるコマンドが記述されています。点灯が不要な場合は、以下の行の mmc0 を none に変更してください。

```
dtoverlay=gpio-led,gpio=22,label=mon_led,trigger=mmc0
```

各ディレクトリ下の brightness に 1 を書くと点灯、0 を書くと消灯します。次のコマンドで MON および ALM が点灯/消灯することを確認してください。

```
pi@raspberrypi:~$ echo 1 | sudo tee /sys/class/leds/mon_led/brightness
pi@raspberrypi:~$ echo 1 | sudo tee /sys/class/leds/alm_led/brightness
pi@raspberrypi:~$ echo 0 | sudo tee /sys/class/leds/mon_led/brightness
pi@raspberrypi:~$ echo 0 | sudo tee /sys/class/leds/alm_led/brightness
```

#### 3.2 プッシュスイッチ

プッシュスイッチは GPIO 経由でソフトウェアから読み取ることができます。差分パッケージ適用後は、プッシュスイッチを押すと Raspberry Pi OS をシャットダウンします。

シャットダウンを行うと、PWR LED の点滅が止まり点灯状態になります。シリアルターミナルにはシャットダウン動作ログが表示され、最後に次のメッセージが表示され OS 動作が止まります。

```
[ 2049.417628] reboot: Power down
```

この状態で安全に電源を切ることができます。

プッシュスイッチによるシャットダウン動作を行わないようにするには、/boot/firmware/config.txt の次の行を削除してから再起動してください。

```
dtoverlay=gpio-shutdown,gpio_pin=6,active_low=1,gpio_pull=up
```

### 3.3 RTC

RTC デバイス(ST m41t93)が SPI 経由で接続されています。差分パッケージ適用後は当該デバイスのドライバが有効になっており、hwclock コマンドで時刻を読み出すことができます。

```
pi@raspberrypi:~$ sudo hwclock
2025-01-20 07:02:06.327763+00:00
```

次のメッセージが出る場合は RTC に時刻が書き込まれていません。

```
hwclock: ioctl(RTC_RD_TIME) to /dev/rtc0 to read the time failed: Invalid argument
```

次のようにして現在時刻を書き込んでください。

```
pi@raspberrypi:~$ sudo date -s "2025/01/20 07:10" ← 現在時刻を UTC で指定
pi@raspberrypi:~$ sudo hwclock -w --noadjfile --utc
pi@raspberrypi:~$ sudo hwclock
2025-01-20 07:10:06.551985+00:00
```

電源 OFF 時に RTC が時刻を保持するにはボタン電池の接続が必要です。ボタン電池の接続については取扱説明書を参照してください。

### 3.4 EEPROM

EEPROM(AT24C02 256 バイト)が I2C 経由で接続されています。差分パッケージ適用後は AT24 ドライバがロードされています。

次のように内容を読み書きできます。

```
pi@raspberrypi:~ $ sudo su
root@raspberrypi:/home/pi# echo ABCDEFGH | sudo tee /sys/bus/i2c/drivers/at24/1-0050/eeprom
root@raspberrypi:/home/pi# hexdump -C /sys/bus/i2c/drivers/at24/1-0050/eeprom
00000000 41 42 43 44 45 46 47 48 0a ff ff ff ff ff ff |ABCDEFGH.....|
00000010 ff |.....|
*
00000100
```

本機出荷時においては EEPROM に特定のデータは書き込まれていません。ユーザーが任意の用途で利用可能です。

### 3.5 イーサネット

#### 3.5.1 USB イーサネット

本機は USB イーサネット LAN9500A を搭載しています。本デバイスは SoM とは USB2.0 で、イーサネットスイッチ KSZ9896 とは MII で接続しています。モードは 100BaseFullDuplex 固定です。差分パッケージ適用後はネットワークインタフェース eth0 として利用できます。

#### 3.5.2 LAN への接続

Raspberry Pi OS 初期状態では DHCP クライアントが稼働しており、LAN 上の DHCP サーバから IP アドレスを取得します。LAN ポート 1/2 いずれかを、DHCP サーバが稼働している LAN へ接続してください。

### 3.5.3 SSH サーバ

Raspberry Pi OS 初期状態では SSH サーバは無効になっています。シリアルコンソールからログインし、`raspi-config` コマンドで SSH サーバを有効化できます。

```
pi@raspberrypi:~$ sudo raspi-config
"3 Interface Options" => "I1 SSH" => "Yes"を選択
```

LAN 上の SSH クライアントから本機へログインできるようになります。

## 3.6 イーサネットスイッチ

本機はギガビットイーサネットスイッチ KSZ9896 を搭載しています。本スイッチは 6 ポートを有しますが本機ではそのうちの 3 ポートを利用しています。LAN ポート 1/2 および USB イーサネット LAN9500A が本スイッチに接続しています。LAN9500A は 100Base での接続となります。

ポート	接続先
1	未使用
2	LAN ポート 1
3	LAN ポート 2
4	未使用
5	未使用
6	SoM(USB イーサネット経由)

また、本スイッチは SoM と SPI で接続されており、本スイッチのレジスタを SoM から制御できます。

### 3.6.1 ポートミラー

本スイッチはポートミラー機能を持っており、LAN ポート 1/2 間のパケットを SoM ヘミラーリングすることができます。

差分パッケージ適用後は `portmirror.service` がインストールされています。このサービスは SPI 経由でスイッチを制御し LAN1/2 間のパケットを SoM ヘミラーリングします。次のコマンドでポートミラーを有効化します。

```
pi@raspberrypi:~ $ sudo systemctl start portmirror
```

起動時からサービスを有効化するには次のコマンドを実行します。

```
pi@raspberrypi:~$ sudo systemctl enable portmirror
```

サービスを無効化する場合は次のコマンドを実行します。

```
pi@raspberrypi:~$ sudo systemctl stop portmirror
pi@raspberrypi:~$ sudo systemctl disable portmirror
```

### 3.6.2 EEE(Energy Efficient Ethernet)の無効化について

本スイッチは、EEE(Energy Efficient Ethernet: IEEE802.3az)を搭載しており機能有効がデフォルト動作となっております。しかしながら、本スイッチの EEE 機能を無効化しようチップベンダより求められており ("KSZ9896 Silicon Errata and Data Sheet Clarification": Item 3)、無効にしない場合の弊害として他の EEE 搭載機器とのリンクドロップを引き起こすことが示されています。

差分パッケージを適用すると、起動時に KSZ9896 の EEE 機能を無効にするコマンドを発行します。以下のコマンドが EEE 機能を無効にするコマンドです。これらコマンドが/etc/rc.local に書かれています。

```
mmd-write 2 0x7 0x3c 0x0000
mmd-write 3 0x7 0x3c 0x0000
```

## 3.7 通信モジュール

### 3.7.1 通信モジュールオプション

本機のオプションとして、以下のモバイル通信モジュールを用意しています。いずれも本機搭載 miniPCIe スロットに接続されます。

通信モジュール型番	説明
SIM7100JC	LTE/Cat3, 3G 対応通信モジュール NTT ドコモ、KDDI(LTE のみ)対応
SIM7100JE	LTE/Cat3, 3G 対応通信モジュール ソフトバンク対応
AK-110	LTE/Cat1, 3G 対応通信モジュール

差分パッケージ適用後は、各通信モジュールを認識し対応ドライバを自動ロードします。モジュールに対応した次のデバイスファイルが作成されます。

通信モジュール	デバイスファイル
SIM7100JC	/dev/ttySIM7100_AT
SIM7100JE	/dev/ttySIM7100_MODEM
AK-110	/dev/ttyAK110_AT /dev/ttyAK110_MODEM

### 3.7.2 SIM カードおよびアンテナの接続

利用する SIM カードを本機のマイクロ SIM カードスロットへ格納してください。また、アンテナを本機アンテナコネクタへ接続してください。SIM カードスロットの場所や、アンテナ接続については本機取扱説明書を参照してください。

### 3.7.3 モバイル通信設定

差分パッケージ適用後は、wvdial を利用して PPP 接続を行う設定ファイルがインストールされます。まず、wvdial をインストールしてください。LAN 経由でインターネットへ接続できる状態で以下のコマンドを実行してください。

```
pi@raspberrypi:~ $ sudo apt-get install wvdial
```

次に、設定ファイルを作成します。

```
pi@raspberrypi:~ $ sudo mv /etc/wvdial_ak130.conf /etc/wvdial.conf
```

次に、設定ファイルを編集し、利用する SIM に対応した設定情報を書き込みます。以下は wvdial.conf の内容を抜粋したものです。通信モジュール毎にセクションが分かれています。利用するモジュールに対応し

たセクションを編集してください。SIM7100 の場合はセクション[Dialer SIM7100]を編集します。  
AK110 の場合はセクション[Dialer AK110]を編集します。

```
...
[Dialer SIM7100]
Init3 = AT+CGDCONT=1,"IP","APN"
Init4 = AT+CGAUTH=1,3,"PASSWORD","USER"
Modem = /dev/ttySIM7100_MODEM
Phone = *99***1#
Password = PASSWORD
Username = USER

[Dialer AK110]
Init3 = AT+CGDCONT=1,"IP","APN"
Init4 = AT$QCPDPP=1,3,"PASSWORD","USER"
Modem = /dev/ttyAK110_MODEM
Phone = *99***1#
Password = PASSWORD
Username = USER
...
```

### APN、ユーザー、パスワードの設定

上記設定ファイル内の APN, USER, PASSWORD を利用する SIM カードに応じて置換してください。

### 認証方式(SIM7100)

SIM7100 を利用する場合、[Dialer SIM7100]セクションの Init4 で始まる行のコマンドを編集してください。2 番目の引数値は認証方式を示します。利用する SIM カードに応じて書き換えてください。

```
Init4 = AT+CGAUTH=1,3,"(パスワード文字列)","(ユーザー文字列)"
      ^ 1=PAP, 2=CHAP, 3=PAP または CHAP
```

### 認証方式(AK110)

AK110 を利用する場合、[Dialer AK110]セクションの Init4 で始まる行のコマンドを編集してください。2 番目の引数値は認証方式を示します。利用する SIM カードに応じて書き換えてください。

```
Init4 = AT$QCPDPP=1,3,"(パスワード文字列)","(ユーザー文字列)"
      ^ 1=PAP, 2=CHAP, 3=PAP または CHAP
```

### 3.7.4 モバイル通信接続

wwan-modem サービスを起動すると wvdial を利用して PPP 接続を行います。サービス名に利用通信モジュール名をつけて実行します。以下は SIM7100 の例です。AK110 の場合は@の後を"AK110"に置き換えてください。

```
pi@raspberrypi:~$ sudo systemctl start wwan-modem@SIM7100
```

起動時に PPP 接続を行う場合は、次を実行します。

```
pi@raspberrypi:~$ sudo systemctl enable wwan-modem@SIM7100
```

PPP 接続が完了すると、ネットワークインタフェース ppp0 が生成されます。

```
pi@raspberrypi:~$ ifconfig
```

```
...
```

```
ppp0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST> mtu 1500
```

```
inet 100.80.11.71 netmask 255.255.255.255 destination 10.64.64.64
ppp txqueuelen 3 (Point-to-Point Protocol)
RX packets 5 bytes 62 (62.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 6 bytes 101 (101.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

...

切断する場合はサービスを停止してください。

```
pi@raspberrypi:~$ sudo systemctl stop wwan-modem@SIM7100
```

## 4. Raspberry Pi OS 公式配布イメージとの差分について

この章では、差分パッケージが Raspberry Pi OS 公式配布イメージに対して行っている変更について示します。Raspberry Pi OS のアップデートや、Raspberry Pi OS 以外の OS を利用する際の参考にしてください。

### 4.1 ツール等

#### 4.1.1 イーサネットスイッチレジスタ操作

`/usr/local/bin/swreg` はイーサネットスイッチ KSZ9896 のレジスタを操作するツールです。`portmirror.service` から使用されます。ソースファイルは `/usr/local/src/swreg` にあります。  
`/usr/local/bin/mmd-read`, `mmd-write` はイーサネットスイッチ KSZ9896 の MMD(MDIO Manageable Device)レジスタを操作するツールです。このコマンドは、EEE(Energy Efficient Ethernet)機能を無効化するために使用します。

### 4.2 設定ファイル等

#### 4.2.1 デバイスツリー

差分パッケージには次のデバイスツリーオーバーレイを追加しています。

オーバーレイファイル名	説明
<code>/boot/firmware/bcm2710-rpi-cm3-ax962.dtb</code>	LED, I2C EEPROM の定義

`dts` ファイルは、`/usr/local/src/dts` 以下にあります。

#### 4.2.2 `/boot/firmware/config.txt`

`/boot/firmware/config.txt` に次の設定行を追加しています。

設定行	説明
<code>kernel=kernel8-ax962.img</code>	Kernel イメージを指定します
<code>device_tree=bcm2710-rpi-cm3-ax962.dtb</code>	デバイスツリーを指定します
<code>hdmi_ignore_hotplug=1</code>	HDMI 検出を無効化し起動を速くします。
<code>dtoverlay=spi1-1cs</code>	RTC が接続されている SPI を有効化します。
<code>dtoverlay=gpio-led,gpio=13,label=pwr_led,trigger=actpwr,active_low</code> <code>dtoverlay=gpio-led,gpio=22,label=mon_led,trigger=mmc0</code> <code>dtoverlay=gpio-</code>	LED を登録します。

led,gpio=23,label=alm_led,trigger=panic	
dtoverlay=gpio-shutdown,gpio_pin=6,active_low=1, gpio_pull=up	プッシュスイッチをシャットダウンボタンとして設定しています。

#### 4.2.3 udev ルール

通信モジュールデバイスドライバのロードおよび tty デバイスにユニーク名を付けるための udev ルールを追加しています。

udev ルールファイル名	説明
/etc/udev/rules.d/99-ak-020.rules	3G ドングル AK020 用
/etc/udev/rules.d/99-ak-110.rules	LTE モジュール AK110 用
/etc/udev/rules.d/99-sim7100.rules	LTE モジュール SIM7100 用
/etc/udev/rules.d/99-abit-lora.rules	Lora モジュール用

#### 4.2.4 systemd サービス

systemd サービスファイルを追加しています。

サービスファイル名	説明
/etc/systemd/system/ipmasq.service	PPP/LAN ルータサンプル
/etc/systemd/system/portmirror.service	イーサネットポートミラー
/etc/systemd/system/wwan-modem@.service	モバイル接続

### 4.3 ソースコード

差分パッケージには GPLv2 ライセンス下のプログラムのバイナリを含んでいます。ソースコードは以下の URL から入手できます。以下の URL へアクセスできない場合は弊社までご連絡ください。

#### Linux

<https://github.com/raspberrypi/linux/>

## 5. 更新履歴

---

版	日付	変更内容
1	2018/03/26	初版
2	2018/04/20	Raspbian 2018-04-18 版対応
3	2018/07/04	Raspbian 2018-06-27 版対応
4	2018/09/07	3.6.2 KSZ9896 の EEE 機能の無効化について記載
5	2018/09/18	2.3 差分パッケージファイル名を訂正
6	2018/10/12	Raspbian 2018-10-09 版対応
7	2018/11/29	Raspbian 2018-11-13 版対応 4.2.3 Lora モジュール用の udev ルールを追加 4.3 GPLv2 プログラムのソースコード情報を追加
8	2019/04/16	Raspbian 2019-04-08 版対応
9	2025/04/28	Raspberry Pi OS 2024-11-20 64-bit 版対応