

AP-RZG-0A

タッチパネル LCD キットの使用方法

Rev1.1 2018/03/09

目次

1. 概要	3
1.1 はじめに.....	3
1.2 開発環境.....	3
2. Linux カーネルへの対応方法	4
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.....	8
3. 動作確認	9
3.1 Linux の起動方法.....	9
3.2 Linux の初回起動、タッチパネルのキャリブレーション.....	10
3.3 ブザーの動作確認.....	12
3.4 バックライトの動作確認.....	13
3.5 タッチスクリーン/ボタンイベントの確認.....	14

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+git34547b2a5032ce6dc  
a24b745d608d2f3baac187f-r0/git 
```

1. 概要

1.1 はじめに

本ドキュメントでは、AP-RZG-0A にタッチパネル LCD キットを接続し、動作確認するまでの手順について説明します。

- ・カーネル対応方法
- ・タッチパネルのキャリブレーション
- ・ブザー、バックライトの動作確認
- ・スクリーンタッチイベント、ボタンイベントの動作確認

1.2 開発環境

タッチパネル LCD キットのドライバを 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。

- ・ Install Manual for RZ/G Series
- ・ LK-RZG-A01 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、タッチパネル LCD キットを使用する設定になっておりませんので、Linux カーネルにドライバを追加する必要があります。

本章ではその手順を説明します。



本手順では、開発環境が、コマンド『**bitbake core-image-x11**』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-A01) 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、タッチパネル LCD キット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 カーネルにタッチパネル LCD キット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 ~/aprzg0a
$ 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~/aprzg0a/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 3.10.31-ltsi Kernel Configuration
Linux/arm 3.10.31-ltsi Kernel Configuration
Arrow keys navigate the menu. <Enter> selects 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 [ ] excluded <M> module < >

[*] 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 --->

v(+)
```

- ③ Touchscreens メニューに移動します。

[Device Drivers] - [Input device support] - [Touchscreens]の順に開いて、『LCD-KIT based touchscreens』を選択します。

その後、下に『select LCD-KIT』が表示されますので、そのリストの中から必要なタッチパネル LCD キットを選択します。

```
.config - Linux/arm 3.10.31-ltsi Kernel Configuration
> Device Drivers > Input device support > Touchscreens
    Touchscreens
    Arrow keys navigate the menu. <Enter> selects 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 [ ] excluded <M> module < >

    --- Touchscreens
    [*] LCD-KIT based touchscreens
       select LCD-KIT (LCD-KIT-B01) --->
    < > ADS7846/TSC2046/AD7873 and AD(S)7843 based touchscreens
    < > AD7877 based touchscreens
    < > Analog Devices AD7879-1/AD7889-1 touchscreen interface
    < > Atmel mXT I2C Touchscreen
    < > AUO in-cell touchscreen using Pixcir ICs
    < > BU21013 based touch panel controllers
    < > cy8ctmg110 touchscreen
    v(+)

    <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『Touchscreens』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 3.10.31-ltsi Kernel Configuration

Do you wish to save your new configuration ? <ESC><ESC>
to continue.

    < 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
    |-- aprzg0a
      |-- build
        |-- tmp
          |-- work
            |-- aprzg0a-poky-linux-gnueabi
              |-- linux-renesas
                |-- 3.10+git34547b2a5032ce6dca24b745d608d2f3baac187f-r0
                  |-- git
                    |-- arch
                      |-- arm
                        |-- boot
                          |-- dts
```

●編集内容

ファイル『r8a7745-aprzg0a.dts』の i2c3 設定部分に各タッチパネル LCD キットが記述されています。該当する箇所のみ『"okay"』とし、それ以外は、『"disabled"』とします。下記例は、LCD-KIT-B01 を使用する場合の設定です。

```
&i2c3 {
    status = "okay";
    clock-frequency = <100000>;

    lcdkitb01: lcdkitb01@0x42 {
        pinctrl-0 = <&lcdkit_pins>;
        pinctrl-names = "default";
        compatible = "apj, lcdkitb01";
        reg = <0x42>;
        interrupt-parent = <&irqc0>;
        interrupts = <9 IRQ_TYPE_LEVEL_HIGH>;

        status = "okay";
    };

    lcdkitc01: lcdkitc01@0x42 {
        pinctrl-0 = <&lcdkit_pins>;
        pinctrl-names = "default";
        compatible = "apj, lcdkitc01";
        reg = <0x42>;
        interrupt-parent = <&irqc0>;
        interrupts = <9 IRQ_TYPE_LEVEL_HIGH>;

        status = "disabled";
    };

    lcdkitd01: lcdkitd01@0x42 {
        pinctrl-0 = <&lcdkit_pins>;
        pinctrl-names = "default";
        compatible = "apj, lcdkitd01";
        reg = <0x42>;
        interrupt-parent = <&irqc0>;
        interrupts = <9 IRQ_TYPE_LEVEL_HIGH>;

        status = "disabled";
    };

    lcdkitd02: lcdkitd02@0x42 {
        pinctrl-0 = <&lcdkit_pins>;
        pinctrl-names = "default";
        compatible = "apj, lcdkitd02";
        reg = <0x42>;
        interrupt-parent = <&irqc0>;
        interrupts = <9 IRQ_TYPE_LEVEL_HIGH>;

        status = "disabled";
    };
};
```

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 ←カ
Loading cache: 100% |#####|
##| ETA: 00:00:00
Loaded 1846 entries from dependency cache.

:
途中省略
:

NOTE: Executing RunQueue Tasks
NOTE: Tasks Summary: Attempted 226 tasks of which 225 didn't need to be rerun and all succeeded.

Summary: There was 1 WARNING message 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 ←カ
Loading cache: 100% |#####|
##| ETA: 00:00:00
Loaded 1846 entries from dependency cache.

:
途中省略
:

NOTE: Tasks Summary: Attempted 233 tasks of which 227 didn't need to be rerun and all succeeded.

Summary: There was 1 WARNING message shown.
```



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-A01) のソフトウェアマニュアルでご確認ください。

3. 動作確認

本章では、LCD、タッチスクリーン、バックライト、ブザーの動作確認方法について説明します。

本章のコマンド入力例では基本的に LCD-KIT-B01 で記載しております。その他のタッチパネル LCD の場合は、適宜読み替えて入力してください。

3.1 Linux の起動方法

タッチパネル LCD キットを追加した 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-A01) のソフトウェアマニュアルでご確認ください。

- ① AP-RZG-0A ボードは、以下のようにタッチパネル LCD キットを接続します。
詳しい接続方法に関しては、使用するタッチパネル LCD キットの『ハードウェアマニュアル』でご確認ください。

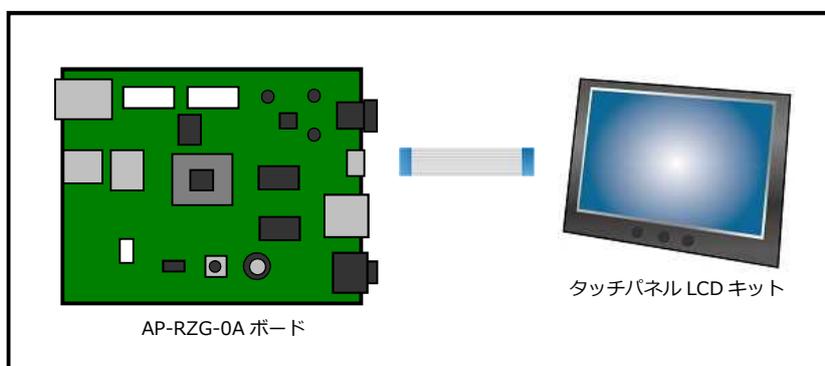


Fig 3.1-1 AP-RZG-0A とタッチパネル LCD キットの接続

- 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。

3.2 Linux の初回起動、タッチパネルのキャリブレーション

起動すると、Linux のペンギンが表示され、その後 Yocto Project の起動画面が表示されます。

初回起動は、自動的にタッチパネルのキャリブレーションプログラムが起動されますが、matchbox のデスクトップ画面の下に隠れてしまうため、正常にキャリブレーションができません。

以下の手順にて、キャリブレーションを実行してください。

- ① matchbox に隠れている、キャリブレーションプログラムの終了を待つ。
xinput_calibrator_once.sh が終了していれば、ps の結果は終了確認のための「grep xinput」だけとなります。
xinput_calibrator_once.sh が実行中の場合はしばらく待ってから再度、確認してください。
(xinput_calibrator_one.sh は、何も操作しないと 15 秒で終了します。)

```
# ps | grep xinput
648 root      2196 S      grep xinput
```

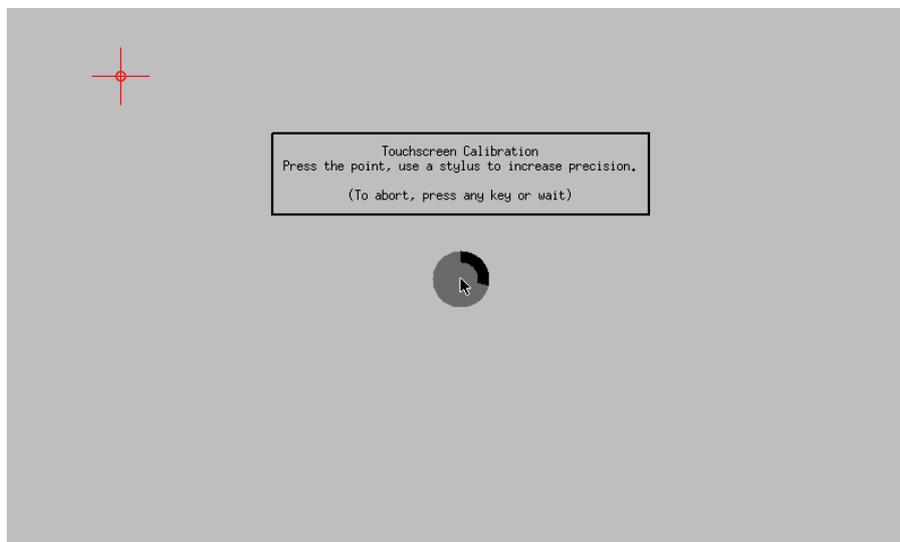
- ② 環境設定をします。

```
# export DISPLAY=0:0
```

- ③ キャリブレーションプログラムを実行します。

```
# xinput_calibrator_once.sh
Calibration data stored in /etc/pointercal.xinput (log in /var/log/xinput_calibrator.poi
ntercal.log)
```

タッチするマークが表示されますので、順番に 4 箇所（左上→右上→左下→右下）をタッチしてください。

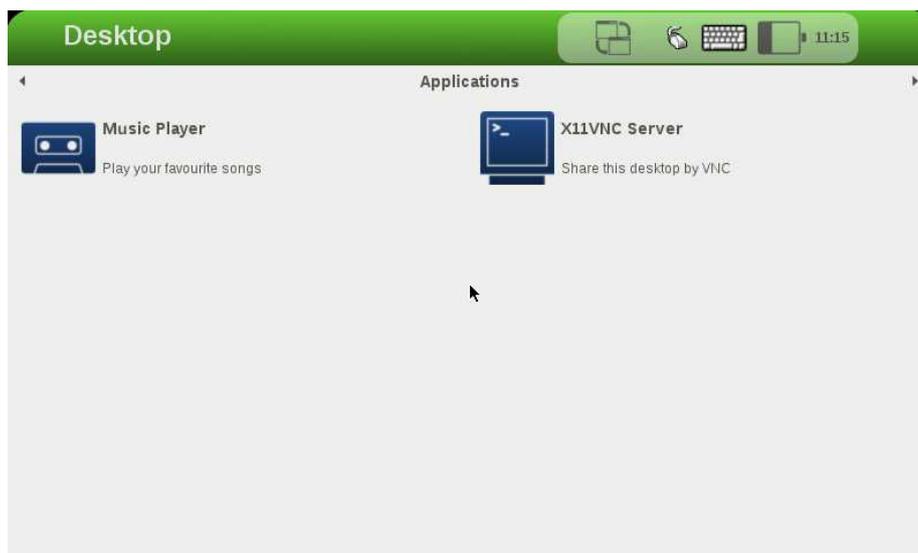


キャリブレーションに失敗したときは『/etc/pointercal.xinput』ファイルを削除し、再度『xinput_calibrator_once.sh』を実行してください。

- ④ 一度、『halt』コマンドで Linux を終了します。

```
# halt   
  
Broadcast message from root@aprzg0a (ttySC10) (Thu Oct 12 09:00:53 2017):  
  
The system is going down for system halt NOW!  
INIT: Sending processes the TERM signal  
INIT * Stopping Avahi mDNS/DNS-SD Daemon: avahi-daemon  
... done.  
Stopping advanced power management daemon: no /usr/sbin/apmd found; none killed  
apmd.  
Stopping system message bus: dbus.  
Stopping syslogd/klogd: stopped syslogd (pid 315)  
stopped klogd (pid 318)  
done  
Stopping Telephony daemon  
ALSA: Storing mixer settings...  
Stopping rpcbind daemon...  
done.  
Stopping Linux NFC daemon  
Deconfiguring network interfaces... ifdown: interface eth0 not configured  
done.  
Sending all processes the TERM signal...  
Sending all processes the KILL signal...  
Unmounting remote filesystems...  
Deactivating swap...  
Unmounting local filesystems...  
System halted.
```

- ⑤ Linux を再起動すると、Linux のペンギンが表示され、その後 Yocto Project の起動画面が表示されます。
最後に、matchbox の Desktop 画面が表示されます。



タッチパネルのキャリブレーションが正しく行われていれば、matchbox の操作が可能となります。

3.3 ブザーの動作確認

タッチパネル LCD キットのブザーは、以下のコマンドで動作確認できます。

- ① タッチパネル LCD キットのブザーを鳴らします。

```
# echo -ne "\x01" > /dev/lcdkitb01_buzzer
```

- ② タッチパネル LCD キットのブザーを止めます。

```
# echo -ne "\x00" > /dev/lcdkitb01_buzzer
```



上記のコマンドで使用しているデバイス名は、LCD-KIT-B01 用となります。
例えば、LCD-KIT-C01 では『**lcdkitc01_buzzer**』となります。

ブザーの鳴動の値に関しては、LCD-KIT-D01 と LCD-KIT-D02 以外は、タッチパネル LCD キットのコマンドの値そのままとなります。

値の詳細に関しては、各タッチパネル LCD キットのハードウェアマニュアルでご確認ください。

LCD-KIT-D01 と LCD-KIT-D02 は、以下となります。

値 (16 進数)	周波数 (Hz)	値 (16 進数)	周波数 (Hz)
0x00	消音	0x10	424.04
0x01	183.75	0x11~0x12	459.38
0x02	190.09	0x13	501.14
0x03	204.17	0x14~0x15	551.25
0x04	212.02	0x16~0x17	612.50
0x05	229.69	0x18~0x1A	689.06
0x06	239.67	0x1B~0x1C	787.50
0x07	250.57	0x1D~0x1F	918.75
0x08	275.63	0x20~0x23	1102.50
0x09	290.13	0x24~0x28	1378.13
0x0A	306.25	0x29~0x2F	1837.50
0x0B	324.26	0x30~0x3B	2756.25
0x0C	344.53	0x3C~0x3F	5512.50
0x0D~0x0E	367.50	0x40~	未対応
0x0F	393.75		

Table 3.4-1 LCD-KIT-D01/D02 ブザー仕様

3.4 バックライトの動作確認

タッチパネル LCD キットのバックライトの制御は、以下のコマンドで確認できます。

- ① タッチパネル LCD キットのバックライト LED を OFF にします。

```
# echo 0 > /sys/class/leds/lcdkitb01_backlight/brightness
```

- ② タッチパネル LED キットのバックライト LED を ON(100%)にします。

```
# echo 255 > /sys/class/leds/lcdkitb01_backlight/brightness
```



バックライト LED の値に関しては、以下の計算後にタッチパネル LCD キットのコマンドとして送信されます。

$$\text{入力値} \times 100 \div 255$$

タッチパネル LCD キットに送る値の詳細は、各タッチパネル LCD のマニュアルでご確認ください。

また、上記のコマンドで使用しているデバイス名は、LCD-KIT-B01 用となります。
例えば、LCD-KIT-C01 では『**lcdkitc01_backlight**』となります。

3.5 タッチスクリーン／ボタンイベントの確認

タッチパネルLCDキットのタッチスクリーン および ボタンイベントは、以下の手順で確認できます。

- ① タッチスクリーン、ボタンイベントを取得します。
終了する場合は、『Ctrl+c』を入力してください。

```
# evtest /dev/input/event0 ←入力
Input driver version is 1.0.1
Input device ID: bus 0x18 vendor 0x0 product 0x0 version 0x0
Input device name: "LCD-KIT-B01"

:
途中省略
:

Max      2048
Testing ... (interrupt to exit)
Event: time 1505966691.663055, type 1 (Key), code 139 (Menu), value 1
Event: time 1505966691.663055, ----- Report Sync -----
Event: time 1505966691.973053, type 1 (Key), code 139 (Menu), value 0
Event: time 1505966691.973053, ----- Report Sync -----
Event: time 1505966693.931060, type 1 (Key), code 102 (Home), value 1
Event: time 1505966693.931060, ----- Report Sync -----
Event: time 1505966694.133050, type 1 (Key), code 102 (Home), value 0
Event: time 1505966694.133050, ----- Report Sync -----
Event: time 1505966695.833043, type 1 (Key), code 1 (Esc), value 1
Event: time 1505966695.833043, ----- Report Sync -----
Event: time 1505966695.993041, type 1 (Key), code 1 (Esc), value 0
Event: time 1505966695.993041, ----- Report Sync -----
```

プッシュ SW1 押した時のイベント

プッシュ SW1 離れた時のイベント

プッシュ SW2 押した時のイベント

プッシュ SW2 離れた時のイベント

プッシュ SW3 押した時のイベント

プッシュ SW3 離れた時のイベント

改定履歴

版数	日付	改定内容
Rev1.0	2017/10/23	新規作成
Rev1.1	2018/03/09	LCD-KIT-D02 対応追加

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。
- ・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。
- ・VirtualBox は、OracleCorporation 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。



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 834
<http://www.apnet.co.jp>
E-MAIL : query@apnet.co.jp