

XG Series

AK-3517-A01

Cortex-A8 AM3517 CPU BOARD

Software Manual

Rev 1.0

ダイジェスト版



ALPHAPROJECT

<http://www.apnet.co.jp>

目次

1. 概要	1
1.1 はじめに	1
1.2 U-Boot について	1
1.3 Android について	1
1.4 VMware Player について	1
1.5 Ubuntu について	2
1.6 GNU と FSF について	2
1.7 GPL と LGPL について	2
1.8 Apache License Version2.0 について	3
1.9 保証とサポート	3
2. システム概要	4
2.1 システム概要	4
2.2 ブートローダ	5
2.3 Android	6
2.4 クロス開発環境	7
2.5 添付 DVD-ROM の構成	8
3. システムの動作	9
3.1 動作環境	9
3.2 シリアル設定	10
3.3 ネットワーク設定	10
3.4 XG-3517 ボードの接続	12
4. ブートローダ	13
4.1 ブートローダの起動	13
4.2 ネットワーク設定	16
4.3 X-Loader の作成	18
4.4 U-Boot の作成	20

5. Android	22
5.1 Android の起動	22
5.2 Linux カーネルの作成	24
5.3 Android の作成	26
5.4 sgx の作成	32
6. microSD カードへのシステム構築	34
6.1 microSD カードの構成.....	34
6.2 作業概要	34
6.3 microSD カードの作成.....	35
6.4 XG-3517 の設定初期化	41
7. FlashROM へのシステム構築	43
7.1 FlashROM 構成	43
7.2 作業概要	43
7.3 Android システム用のデータ作成	44
7.4 microSD カードの作成.....	47
7.5 FlashROM への書き込み.....	51
8. 製品サポートのご案内	55
9. エンジニアリングサービスのご案内	56
付録 A. 付属品について	57

2. システム概要

2.1 システム概要

XG-3517 は、CPU コアに ARM Cortex-A8 を採用したマイクロプロセッサ「AM3517」(TEXAS INSTRUMENTS)を搭載した汎用 CPU ボードです。

Android システムは、ブートローダと Linux カーネル、Android から構成されます。ブートローダに X-Loader と U-Boot、Linux カーネルに Linux3.1、Android にはバージョン 2.3 を使用します。

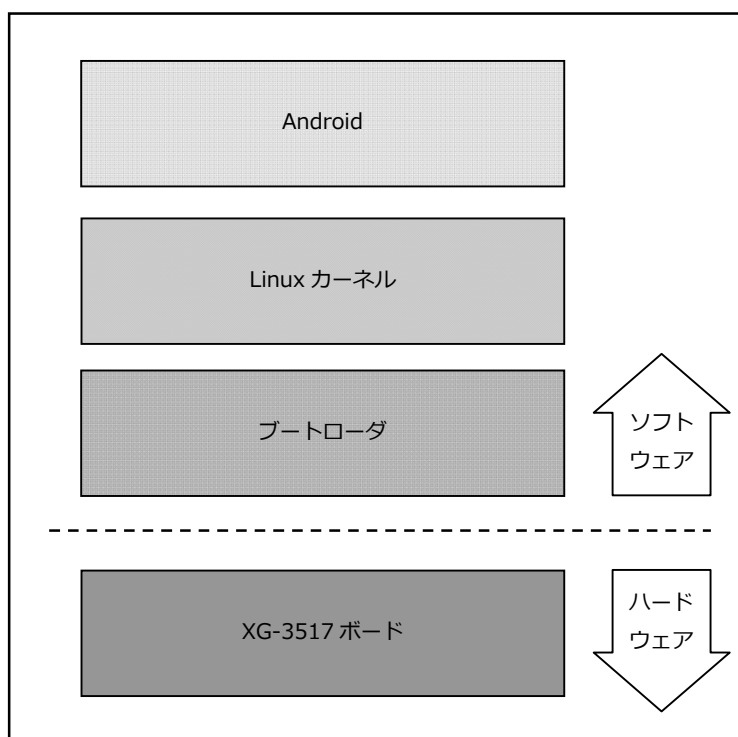


Fig 2.1-1 XG-3517 システム概要図

2.3 Android

Android は、カーネル、ミドルウェア、ユーザーインターフェース、Web ブラウザ等のソフトウェアを 1 つのパッケージにして提供されています。

Android ランタイムには、Dalvik VM(Dalvik 仮想マシン)が存在し、Android 用のアプリケーションは、基本的にこの DalvikVM 上で動作し、プレインストールアプリケーションと後からインストールするアプリケーションを区別なく扱います。

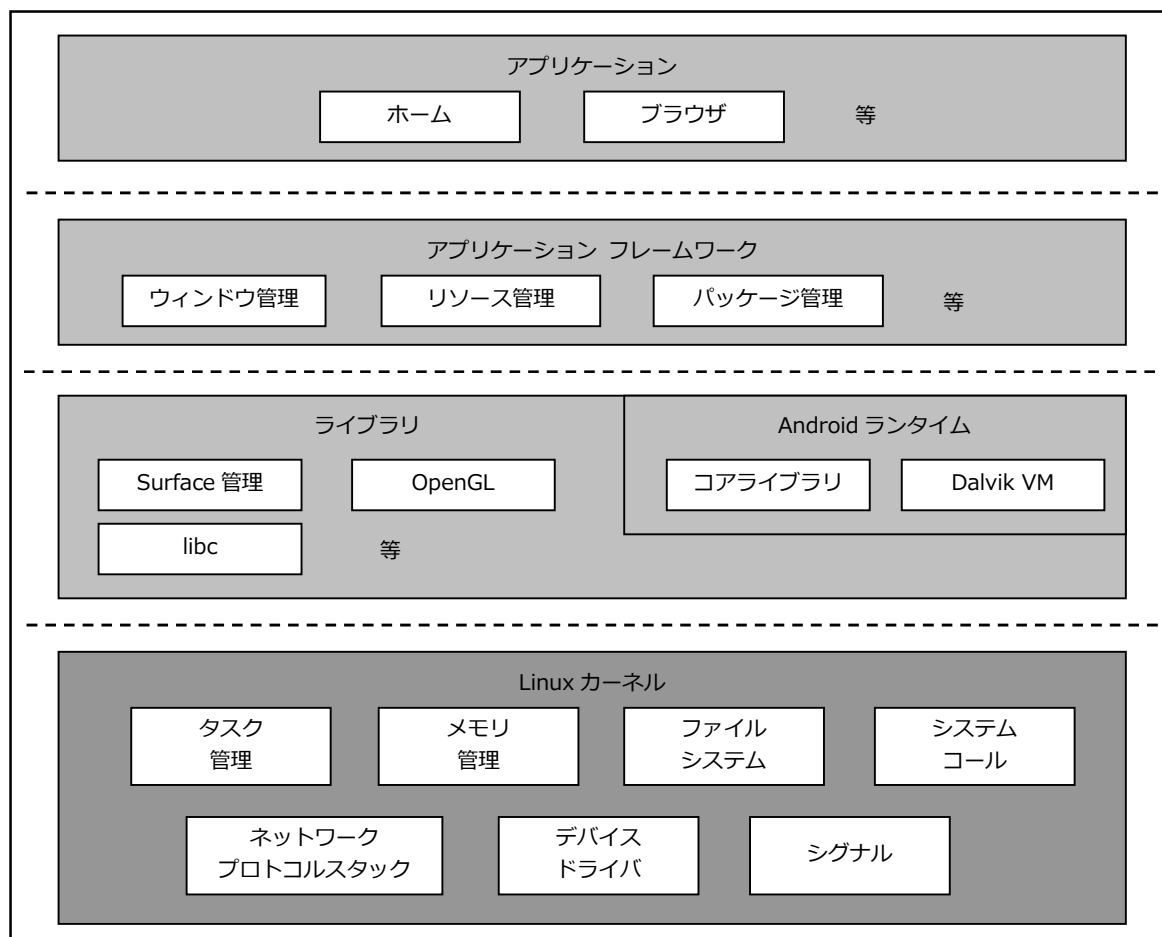


Fig 2.3-1 Android システム概要

2.5 添付 DVD-ROM の構成

XG-3517 の Android の開発には、Linux カーネルソース、Android ソースファイル、クロスコンパイラ等が必要です。これらは、弊社ホームページ及び関連リンクからダウンロードするか、添付 DVD-ROM から入手することができます。

AK_3517_A01_VX_X	
-- binaries	
-- MLO	: X-Loader バイナリ
-- rootfs-android-xg3517.img	: Android ファイルシステム一式 (FlashROM 用)
-- rootfs-android-xg3517.tar.bz2	: Android ファイルシステム一式 (SD 用)
-- u-boot.bin	: U-Boot バイナリ
`-- uImage-xg3517	: Linux カーネルバイナリ
-- index.html	: インデックス HTML
-- index_images	: インデックス HTML イメージ
-- jtag_cnv_01	: JTAG-CNV-01 関連一式
-- license	
-- fdl.txt	: GFDL 原文
-- gpl.txt	: GPL 原文
-- lgpl.txt	: LGPL 原文
`-- LICENSE-2.0.txt	: Apache License Version2.0 原文
-- manual	
-- ak_3517_a01_sw.pdf	: AK-3517-A01 ソフトウェアマニュアル
-- ak_install.pdf	: Android 開発 インストールマニュアル
`-- ere08.pdf	: 未実装の機能について
-- pc_usb_03	: PC-USB-03 関連一式
-- sources	
-- android_kit-xg3517-X.X.tar.bz2	: Android ソースファイル
-- hardware-ti-sgx-xg3517-X.X.tar.bz2	: sgx ファイル一式
-- kernel-ce1b928-xg3517-android-X.X.tar.bz2	: Linux カーネルソースファイル
-- u-boot-2011.09-xg3517-android-X.X.tar.bz2	: U-Boot ソースファイル
`-- xloader-xg3517-X.X.tar.bz2	: X-Loader ソースファイル
`-- toolchain	
-- android-toolchain-eabi.tar.bz2	: クロスコンパイラ一式 (Android 用)
-- arm-cross-x86_64.tar.bz2	: クロスコンパイラ一式 (U-Boot 等用)
`-- install-toolchain.sh	: インストールスクリプト

Table 2.5-1 DVD-ROM 内容

※『X_X』、『X.X』はバージョン番号を示します。バージョン 1.0 の場合は『1_0』、『1.0』になります。

3. システムの動作

3.1 動作環境

Android の起動を確認するためには、CPU ボードと以下の環境が必要です。

●ホスト PC

XG-3517 の開発では、PC をコンソール端末としても使用します。

本 Android 開発キットには、PC-USB-03 が付属しており、PC-USB-03 と PC を USB ケーブルにより接続することで、PC 上では仮想シリアルポートとして認識します。

PC-USB-03 の使用方法に関しては、PC-USB-03 のマニュアルをご参照ください。

なお、仮想シリアルポートを使用した通信には、ハイパーターミナル等のターミナルソフトウェアが別途必要となります。

使用機器等	環 境
CPU ボード	XG-3517
HOST PC	PC/AT 互換機
OS	WindowsXP/Vista/7
メモリ	使用 OS による
ソフトウェア	ターミナルソフト
USB ポート	1 ポート
LAN ポート	10/100BASE-TX 1 ポート
SD カードスロット	microSD カードが読み込めるスロット(Ubuntu から認識できること)
LCD-KIT-B01	ユーザインターフェースとして使用
PC-USB-03	ホスト PC と XG-3517 のシリアル接続用に使用
USB ケーブル	PC-USB-03 で使用
LAN ケーブル	ホスト PC と接続時はクロスケーブルを使用 ハブと接続時はストレートケーブルを使用
電源	AC アダプタ (DC5V±5%) LCD-KIT-B01 を使用するため、2A 以上必要

Table 3.1-1 動作環境



上記の環境は、XG-3517 の Android の動作確認をするための環境となります。

Android 等のコンパイルに使用する開発環境に関しては、『**Android 開発 インストールマニュアル**』でご確認ください。

3.4 XG-3517 ボードの接続

ホスト PC、XG-3517 ボード、LCD-KIT-B01 の接続例を示します。

LAN をネットワークと接続する場合は、ネットワーク管理者と相談し、設定に注意して接続してください。



LCD-KIT-B01 の接続に関しては、『LCD-KIT-B01 ハードウェアマニュアル』でご確認ください。

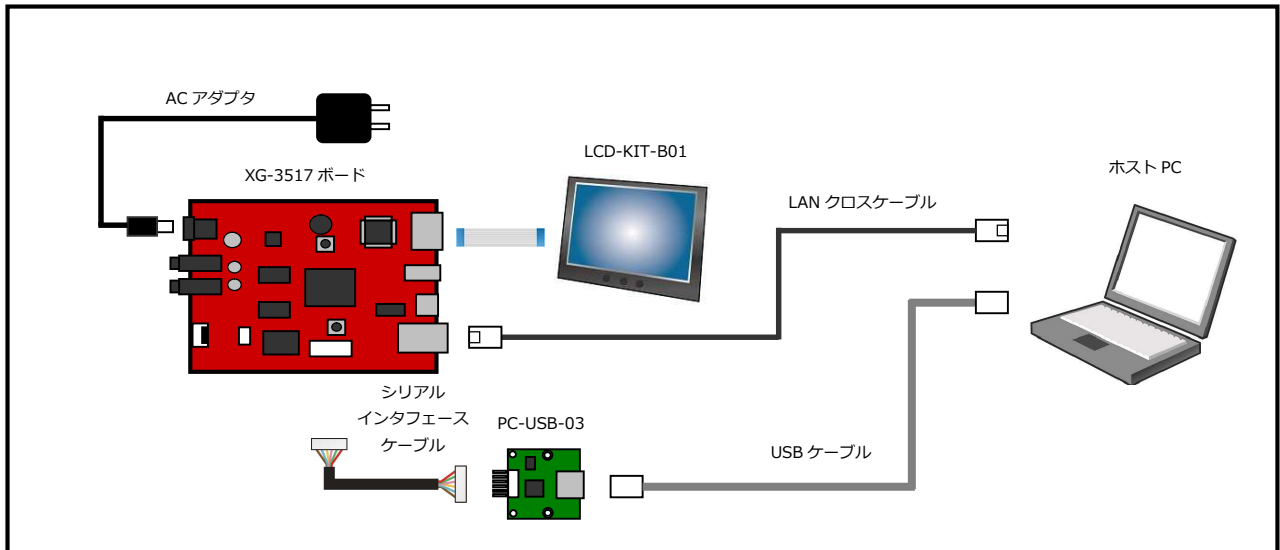


Fig 3.4-1 XG-3517 ボードの接続 (PC に接続する場合)

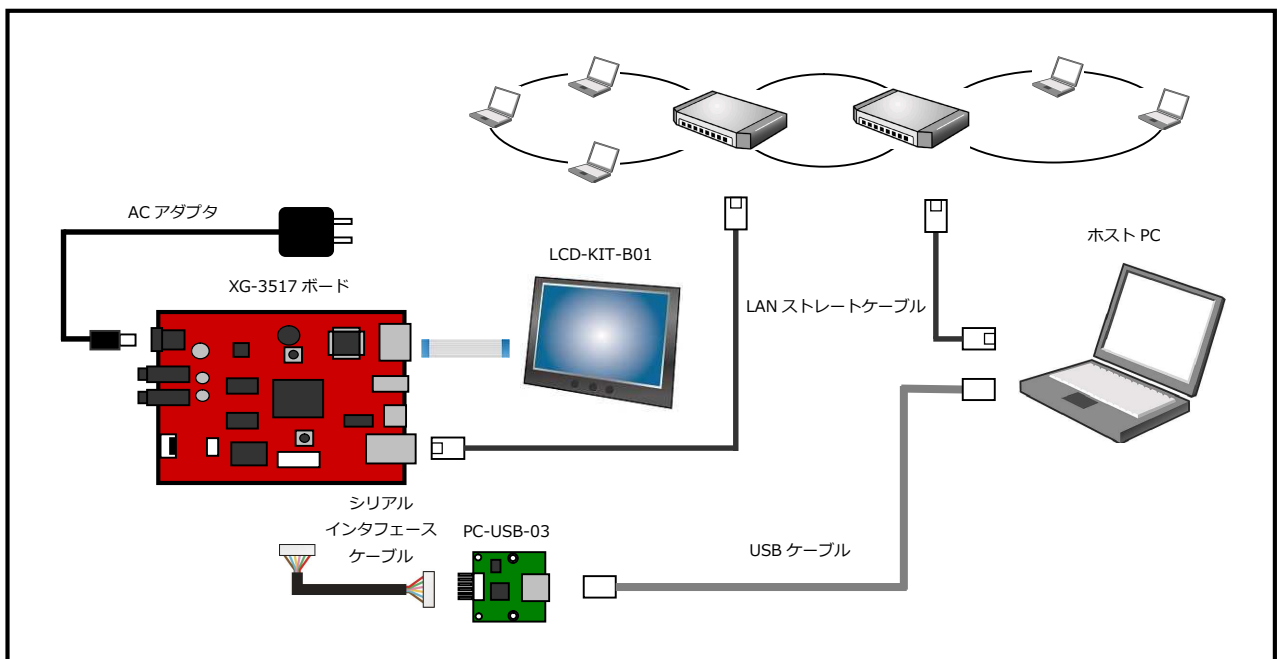


Fig 3.4-2 XG-3517 ボードの接続 (HUB に接続する場合)

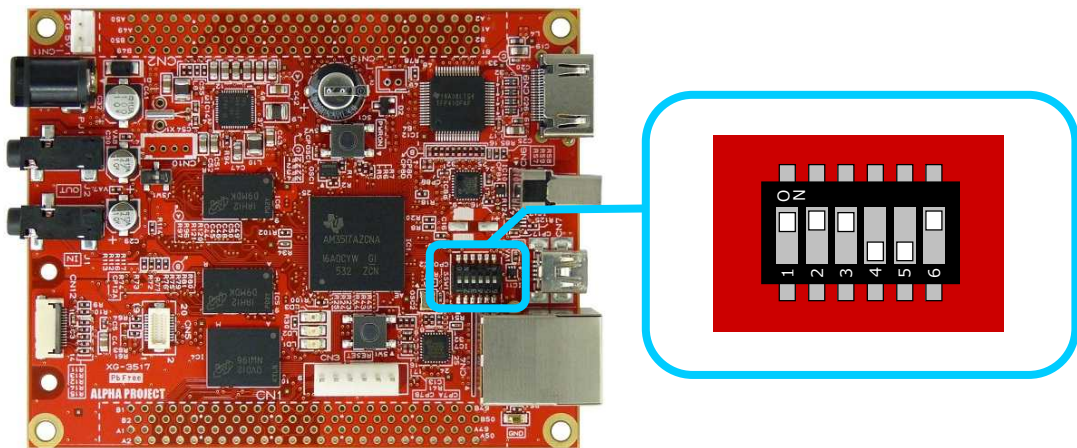
4. ブートローダ

4.1 ブートローダの起動

XG-3517 を起動して、X-Loader が U-Boot をロード後、U-Boot のコマンドコンソールに入る方法を説明します。
なお、起動方法には、microSD カードからの起動と FlashROM からの起動の 2 つ方法がありますので、それぞれについて記述します。

4.1.1 microSD カードからの起動

- ① Android システムを構築した microSD カードを XG-3517 のスロットに挿入します。
microSD カードの作成方法は、『[6. microSD カードへのシステム構築](#)』でご確認ください。
- ② XG-3517 のディップスイッチが、以下のようになっていることを確認します。
ディップスイッチの各設定の詳細に関しては、『[XG-3517 ハードウェアマニュアル](#)』でご確認ください。



- ③ 『[3.4 XG-3517 ボードの接続](#)』にしたがって、ホスト PC、XG-3517、LCD-KIT-B01 を接続します。
- ④ AC アダプタを接続して、XG-3517 の電源を入れます。
PC-USB-03 が、ホスト PC に認識されて仮想 COM ポートが作成されます。
- ⑤ ホスト OS (Windows) のターミナルソフトを起動します。(設定は『[3.2 シリアル設定](#)』を参照してください)
- ⑥ XG-3517 のリセットスイッチ(SW1)を押して、ボードをリセットします。

4.4 U-Boot の作成

ゲスト OS(Ubuntu)上で U-Boot をコンパイルするための手順を説明します。

U-Boot 作成の準備

- ① 作業用ディレクトリ『xg3517-ak』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。

```
省略 $ mkdir ~/xg3517-ak
```

- ② 手順①で作成した作業用ディレクトリに移動します。

```
省略 $ cd ~/xg3517-ak
```

- ③ 作業用ディレクトリに付属 DVD 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 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。

u-boot-2011.09-xg3517-android-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 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。

```
省略 $ 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 のボリュームラベルになります。)

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

- ⑤ ファイルをコピーします。コマンド途中の『*****』は、DVD 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 に合わせて入力してください。

```
省略 $ cp /media/*****/sources/u-boot-2011.09-xg3517-android-X.X.tar.bz2 .
```

- ⑥ DVD をアンマウントします。

```
省略 $ umount /dev/sr0
```

- ⑦ ソースファイルを展開します。

```
省略 $ tar -xjpf u-boot-2011.09-xg3517-android-X.X.tar.bz2
```

U-Boot の作成

- ① 準備作業で展開した作業用ディレクトリの『u-boot-2011.09-xg3517-android-X.X』へ移動します。

```
省略 $ cd ~/xg3517-ak/u-boot-2011.09-xg3517-android-X.X
```

- ② XG-3517 用のコンフィグレーションをします。

```
省略 $ make xg3517_config  
awk '(NF && $1 !~ /^#/)' { print $1 ": " $1 "_config; $(MAKE)" }' boards.cfg > .boards.depend  
Configuring for xg3517 board..
```

- ③ コンパイルします。

```
省略 $ make  
Generating include/autoconf.mk  
Generating include/autoconf.mk.dep  
  
:  
途中省略  
:  
  
arm-linux-objcopy -O srec u-boot u-boot.srec  
arm-linux-objcopy --gap-fill=0xff -O binary u-boot u-boot.bin
```

- ④ make が正常に終了するとカレントディレクトリに『u-boot.bin』が作成されます。

```
省略 $ ls u-boot.bin  
u-boot.bin
```

5. Android

5.1 Android の起動

Android の起動方法を説明します。

起動方法には、microSD カードからの起動と FlashROM からの起動の 2 つ方法がありますので、それぞれについて記述します。

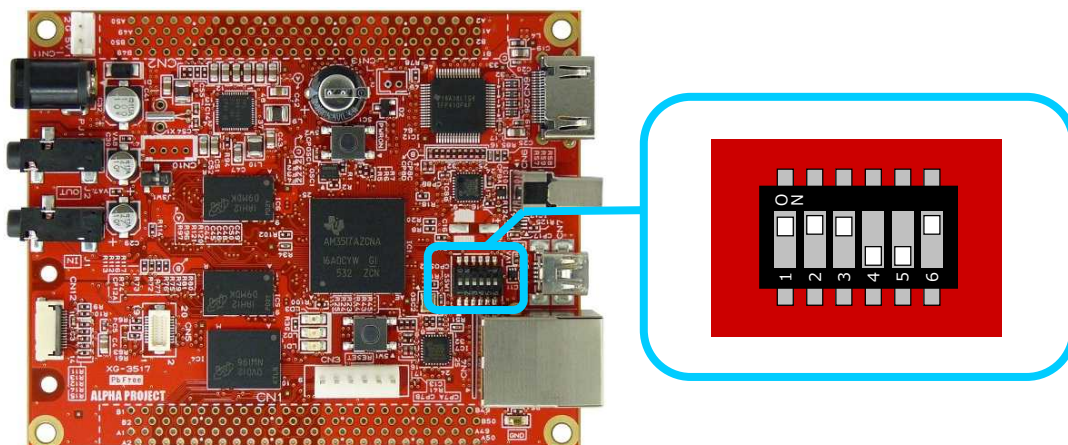
なお、Power スイッチ(SW2)の動作に関しては、起動方法に関係なく以下の動作となります。

XG-3517 の状態	動作内容
起動中	1 回押し : サスペンドに移行 長押し : 電源切るメニュー表示
サスペンド中	サスペンドを解除

Table 5.1-1 Power スイッチの動作

5.1.1 microSD カードからの起動

- Android システムを構築した microSD カードを XG-3517 のスロットに挿入します。
microSD カードの作成方法は、『[6. microSD カードへのシステム構築](#)』でご確認ください。
- XG-3517 のディップスイッチが以下の設定になっていることを確認します。
ディップスイッチの各設定の詳細に関しては、『[XG-3517 ハードウェアマニュアル](#)』でご確認ください。



- 『[3.4 XG-3517 ボードの接続](#)』にしたがって、ホスト PC、XG-3517、LCD-KIT-B01 を接続します。
(ホスト PC と XG-3517 の接続は、必須ではありません。必要に応じて接続してください。)
- XG-3517 の電源を入れます。
AC アダプタを接続すると XG-3517 の電源が入ります。電源が投入されると、Android が起動します。
起動が完了しますと、LCD-KIT-B01 の LCD に Android のホーム画面が表示されます。

Linux カーネルの作成

Linux カーネルをコンパイルする方法を説明します。

Linux カーネルの設定データは、カーネルソースディレクトリ以下『arch/arm/configs/xg3517_android_defconfig』に保存されています。

- ① 準備作業で展開した作業用ディレクトリの『kernel-ce1b928-xg3517-android-X.X』へ移動します。

```
省略 $ cd ~/xg3517-ak/kernel-ce1b928-xg3517-android-X.X ←入力
```

- ② Linux カーネルの設定データを呼び出します。

```
省略 $ make xg3517_android_defconfig ←入力
HOSTCC scripts/basic/fixdep
HOSTCC scripts/kconfig/conf.o
SHIPPED scripts/kconfig/zconf.tab.c
SHIPPED scripts/kconfig/zconf.lex.c
SHIPPED scripts/kconfig/zconf.hash.c
HOSTCC scripts/kconfig/zconf.tab.o
HOSTLD scripts/kconfig/conf
#
# configuration written to .config
#
```

- ③ make を実行します。終了までに数分から数時間かかる場合があります。

```
省略 $ make uImage ←入力
scripts/kconfig/conf --silentoldconfig Kconfig
WRAP arch/arm/include/generated/asm/auxvec.h
WRAP arch/arm/include/generated/asm/bitperlong.h
:
途中省略
:
Load Address: 0x80008000
Entry Point: 0x80008000
Image arch/arm/boot/uImage is ready
```

- ④ make が正常に終了すると『./arch/arm/boot』ディレクトリに Linux カーネルイメージ『uImage』が作成されます。

```
省略 $ ls arch/arm/boot/uImage ←入力
arch/arm/boot/uImage
```

5.3 Android の作成

Android 作成の準備

- ① 作業用ディレクトリ『xg3517-ak』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。

```
省略 $ mkdir ~/xg3517-ak
```

- ② 手順①で作成した作業用ディレクトリに移動します。

```
省略 $ cd ~/xg3517-ak
```

- ③ 作業用ディレクトリに付属 DVD 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 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。

```
android_kit-xg3517-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 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。

```
省略 $ 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 のボリュームラベルになります。)

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

- ⑤ 2 つのファイルをコピーします。コマンド途中の『*****』は、DVD 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 に合わせて入力してください。

```
省略 $ cp /media/*****/sources/android_kit-xg3517-X.X.tar.bz2 .
```

- ⑥ DVD をアンマウントします。

```
省略 $ umount /dev/sr0
```

- ⑦ 2 つのソースファイルを展開します。

```
省略 $ tar -xjpf android_kit-xg3517-X.X.tar.bz2
```

ギャラリー保存領域の設定

Android のギャラリー保存領域は、デフォルトで microSD カードの第 3 パーティションに設定されています。
この保存領域を XG-3517 の CN6 に挿入した USB メモリに変更する方法を記述します。

Android 作成の準備で展開したソースファイルの以下の 3 つのファイルを変更します。

- ① マウントする領域を変更するため、『**void.fstab**』ファイルを変更します。
15 行目の『**dev_mount...**』をコメント行にし、16 行目の『**dev_mount...**』のコメントを解除します。

対象ファイルパス

android_kit-xg3517-X. X/device/alphaproject/xg3517_android/void.fstab

変更前

```
      :  
      省略  
      :  
  
dev_mount sdcard /mnt/sdcard 3 /devices/platform/omap/omap_hsmmc.0/mmc_host/mmc0  
# dev_mount sdcard /mnt/sdcard auto /devices/platform/usbhs-omap.0/ehci-omap.0/usb1
```

変更後

```
      :  
      省略  
      :  
  
# dev_mount sdcard /mnt/sdcard 3 /devices/platform/omap/omap_hsmmc.0/mmc_host/mmc0  
dev_mount sdcard /mnt/sdcard auto /devices/platform/usbhs-omap.0/ehci-omap.0/usb1
```

6. microSD カードへのシステム構築

本章では、添付されている microSD カードに Android システムを構築する手順を説明します。

6.1 microSD カードの構成

本章の手順に従って microSD カードに Android システムを構築した場合は、以下の構成となります。
各パーティションの容量に関しては、必要に応じて変更してください。

microSD カード 4GByte		
フォーマット	使用用途	領域サイズ
FAT32	X-Loader 等の格納領域	約 156MByte
EXT3	Android 等の格納領域	約 3GByte
FAT32	ギャラリーの保存領域	約 658MByte

Fig 6.1-1 microSD カード構成

6.2 作業概要

システム構築手順は、以下の 2 つの作業で行います。

1. MMC ブート用の microSD カードを作成します。
2. XG-3517 に保存されている U-Boot の環境変数領域を初期化します。

次節より、上記の作業順番で説明します。

6.3 microSD カードの作成

作成の準備

- ① 作業用ディレクトリ『**xg3517-ak**』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。

```
省略 $ mkdir ~/xg3517-ak
```

- ② 本手順で使用するデータの置く場所用にディレクトリ『**mmc**』を作業用ディレクトリに作成します。

```
省略 $ mkdir ~/xg3517-ak/mmc
```

- ③ 手順②で作成したディレクトリに移動します。

```
省略 $ cd ~/xg3517-ak/mmc
```

- ④ 作成するためには、以下の4つのファイルが必要となります。

MLO
u-boot.bin
uImage-xg3517
rootfs-android-xg3517.tar.bz2

本ドキュメント内で作成したファイルでも可能ですが、同等のファイルがDVD内にあります。
以下の手順⑤～⑦では、DVDからコピーする方法で説明します。



rootfs-android-xg3517.tar.bz2 は、『[5.3 Android の作成](#)』及び『[5.4 sgx の作成](#)』で作成したファイルを圧縮したファイルとなります。本ドキュメントで作成したファイルを使用する場合には、このファイルは必要ありません。

- ⑤ DVD 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。

```
省略 $ gvfs-mount -d /dev/sr0 ←入力
```



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 のボリュームラベルになります。)

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

- ⑥ ファイルをコピーします。コマンド途中の『*****』は、DVD 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 に合わせて入力してください。

```
省略 $ cp /media/*****/binaries/ML0 . ←入力
省略 $ cp /media/*****/binaries/u-boot.bin . ←入力
省略 $ cp /media/*****/binaries/uImage-xg3517 . ←入力
省略 $ cp /media/*****/binaries/rootfs-android-xg3517.tar.bz2 . ←入力
```

- ⑦ DVD をアンマウントします。

```
省略 $ umount /dev/sr0 ←入力
```

作成

- ① microSD カードをホスト PC の SD カードスロットに挿入して、Ubuntu 上で操作できるようにします。



Ubuntu で microSD カードを認識した場合、自動でマウントされる場合があります。

その場合には、すべてアンマウントしてから行うようにしてください。

- ② 準備手順で用意したデータがある場所に移動します。

```
省略 $ cd ~/xg3517-ak/mmc ←入力
```

- ③ 『sudo fdisk /dev/sdb』を実行します。

```
省略 $ sudo fdisk /dev/sdb ←入力
[sudo] password for guest: ←入力
```

警告: DOS互換モードは廃止予定です。このモード(コマンド 'c')を止めることを強く推奨します。and change display units to sectors (command 'u').

コマンド (m でヘルプ):

- ④ 領域を全て削除します。

```
コマンド (m でヘルプ) : d ←入力
選択した領域 1
```

- ⑤ microSD カードの情報を取得します。

```
コマンド (m でヘルプ) : p ←入力

ディスク /dev/sdb: 3965 MB, 3965190144 バイト
ヘッド 49, セクタ 48, シリンダ 3292
Units = シリンダ数 of 2352 * 512 = 1204224 バイト
セクタサイズ (論理 / 物理) : 512 バイト / 512 バイト
I/O size (minimum/optimal) : 512 bytes / 512 bytes
ディスク識別子: 0x00000000

デバイス ブート 始点 終点 ブロック Id システム
```



以降の作業でシリンダ数の入力があります。その時の値は、ディスクの総容量(上記ログの『3965190144 バイト』)から計算した値を設定する必要があります。
以降の説明では、上記サイズでの説明となりますので、異なる場合にはご注意ください。

- ⑥ 手順⑤で取得したデータよりシリンダ数を変更します。

```
コマンド (m でヘルプ) : x ←入力

上級者コマンド (m でヘルプ) : h ←入力
ヘッド数 (1-256, 初期値 110) : 255 ←入力

上級者コマンド (m でヘルプ) : s ←入力
セクタ数 (1-63, 初期値 45) : 63 ←入力
警告: DOS 互換のためのセクタオフセットを設定します

上級者コマンド (m でヘルプ) : c ←入力
シリンダ数 (1-1048576, 初期値 1627) : 482 ←入力

上級者コマンド (m でヘルプ) : r ←入力
```



シリンダ数は、ディスク容量に応じて以下の計算から値を決定します。
全容量 ÷ 255 ÷ 63 ÷ 512

そのため、上記の手順では、以下の計算となり、482 となります。
 $3965190144 \div 255 \div 63 \div 512 = 482$ (小数点以下切捨て)

7. FlashROM へのシステム構築

本章では、作成した Android システムを FlashROM に構築する手順を説明します。

7.1 FlashROM 構成

本章の手順に従って microSD カードに Android システムを構築した場合は、以下の構成となります。各パーティションの容量に関しては、必要に応じて変更してください。

FlashROM 512MByte		
開始アドレス	領域名	領域サイズ
0x00000000	X-Loader	128KByte
0x00020000	U-Boot	384KByte
0x00080000	U-Boot 環境変数領域	128KByte
0x000A0000	未使用	384KByte
0x00100000	Linux カーネル	6MByte
0x00700000	未使用	6MByte
0x00D00000	Android システム	499MByte

Fig 7.1-1 FlashROM 構成

7.2 作業概要

FlashROM に書き込む手順は、以下の 3 つの作業で行います。

1. 作成した Android システムのデータを FlashROM に書き込めるように準備します。
2. MMC ブート用の microSD カードを作成します。
microSD カードを MMC ブートできるように作成し、FlashROM に書き込むデータも入れます。
(ただし、1 で準備したデータは、TFTP 転送できるように Ubuntu 側におきます。)
3. FlashROM に用意したデータを書き込みます。
MMC ブートにより起動した U-Boot を利用して、FlashROM にデータを書き込みます。

次節より、上記の作業順番で説明します。

7.4 microSD カードの作成

FlashROM に書き込むための microSD カードを準備します。

作成の準備

- ① 作業用ディレクトリ『**xg3517-ak**』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。

```
省略 $ mkdir ~/xg3517-ak ←入力
```

- ② 本手順で使用するデータの置く場所用にディレクトリ『**flash**』を作業用ディレクトリに作成します。

```
省略 $ mkdir ~/xg3517-ak/flash ←入力
```

- ③ 手順②で作成したディレクトリに移動します。

```
省略 $ cd ~/xg3517-ak/flash ←入力
```

- ④ 書き込む作業には、以下の4つのファイルが必要となります。

```
MLO
u-boot.bin
uImage-xg3517
```

それぞれ『[4.3 X-Loader の作成](#)』、『[4.4 U-Boot の作成](#)』、『[5.2 Linux カーネルの作成](#)』で作成したファイルでも可能ですが、同等のファイルが DVD 内にもあります。

以下の手順⑤～⑦では、DVD からコピーする方法で説明します。

- ⑤ DVD 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。

```
省略 $ gvfs-mount -d /dev/sr0 ←入力
```



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 のボリュームラベルになります。)

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

- ⑥ ファイルをコピーします。コマンド途中の『*****』は、DVD 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 に合わせて入力してください。

```
省略 $ cp /media/*****/binaries/ML0 .
省略 $ cp /media/*****/binaries/u-boot.bin .
省略 $ cp /media/*****/binaries/uImage-xg3517 .
```

- ⑦ DVD をアンマウントします。

```
省略 $ umount /dev/sr0
```

作成

- ① microSD カードをホスト PC の SD カードスロットに挿入して、Ubuntu 上で操作できるようにします。



Ubuntu で microSD カードを認識した場合、自動でマウントされる場合があります。
その場合には、すべてアンマウントしてから行うようにしてください。

- ② 準備手順で用意したデータがある場所に移動します。

```
省略 $ cd ~/xg3517-ak/flash
```

- ③ 『`sudo fdisk /dev/sdb`』を実行します。

```
省略 $ sudo fdisk /dev/sdb
[sudo] password for guest:
```

```
警告: DOS互換モードは廃止予定です。このモード(コマンド'c')を止めることを
強く推奨します。and change display units to
sectors (command'u').
```

コマンド (m でヘルプ):

- ④ 領域を全て削除します。

```
コマンド (m でヘルプ): d
選択した領域 1
```

- ⑤ microSD カードの情報を取得します。

```
コマンド (m でヘルプ): p
```

```
ディスク /dev/sdb: 3965 MB, 3965190144 バイト
ヘッド 49, セクタ 48, シリンダ 3292
Units = シリンダ数 of 2352 * 512 = 1204224 バイト
セクタサイズ (論理 / 物理): 512 バイト / 512 バイト
I/O size (minimum/optimal): 512 bytes / 512 bytes
ディスク識別子: 0x00000000
```

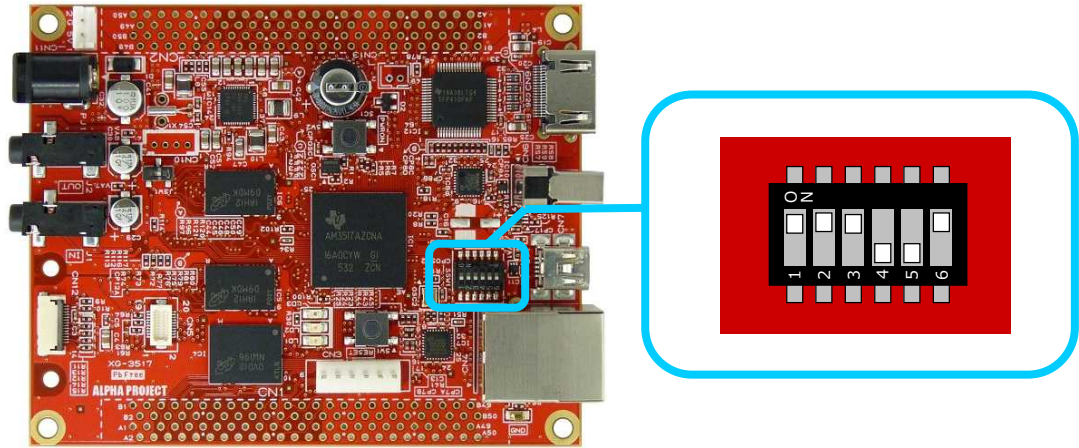
```
デバイス ブート 始点 終点 ブロック Id システム
```



以降の作業でシリンダ数の入力があります。その時の値は、ディスクの総容量(上記ログの『3965190144 バイト』)から計算した値を設定する必要があります。
以降の説明では、上記サイズでの説明となりますので、異なる場合にはご注意ください。

7.5 FlashROM への書き込み

- ① XG-3517 のディップスイッチを以下の設定に変更します。
ディップスイッチの各設定の詳細に関しては、『XG-3517 ハードウェアマニュアル』でご確認ください。



- ② 下図に従って、作成した microSD カードを XG-3517 に挿入し、PC-USB-03 を使用して PC と XG-3517 のシリアルポート (CN3) を接続します。

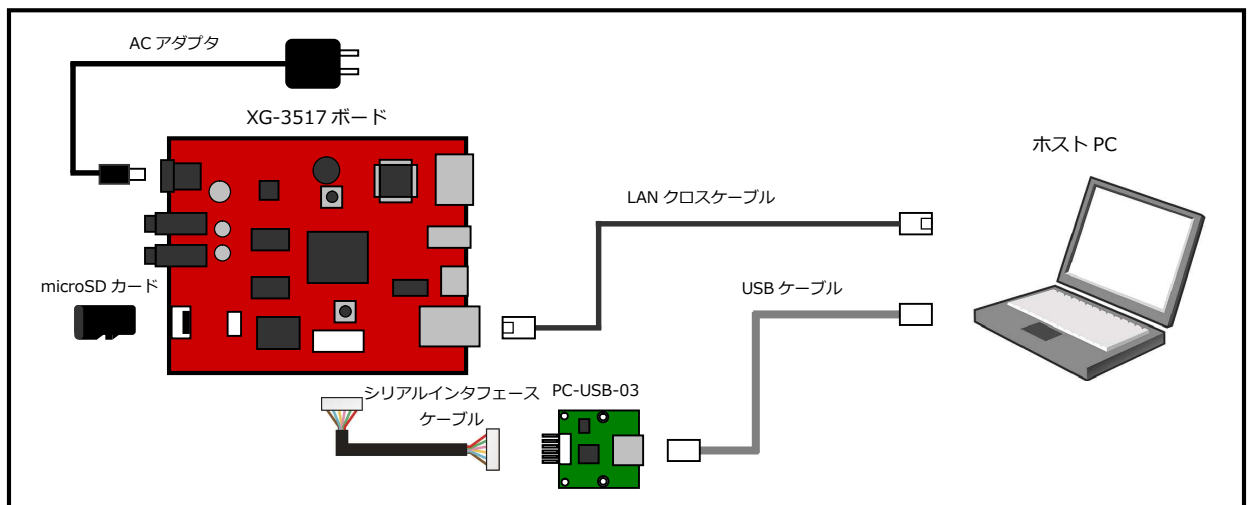


Fig 7.5-1 ボードの接続

※上記は、ネットワークに LAN クロスケーブルを使用した場合の接続になります。HUB 等を使用したい場合には、『3.4 XG-3517 ボードの接続』を参考に接続してください。

- ③ XG-3517 の電源を入れます。
PC-USB-03 が PC に認識されて仮想 COM ポートが作成されます。

- ④ ホスト OS (Windows) のターミナルソフトを起動します。(設定は『[3.2 シリアル設定](#)』を参照してください)
- ⑤ XG-3517 のリセットスイッチ(SW1)を押して、ボードをリセットします。
- ⑥ リセット後、ターミナルに、『Hit any key to stop autoboot』の文字が表示され、2 秒以内にキー入力を行うと U-Boot のコマンドコンソールが起動します。
コマンドコンソールが起動すると、『=>』が表示されます。

```
U-Boot 2011.09 (Jul 30 2012 - 16:16:14) ALPHAPROJECT XG-3517 Android vX.X

AM35XX-GP ES2.0, CPU-OPP2, L3-165MHz, Max CPU Clock 600 Mhz
XG-3517 + LPDDR/NAND
I2C:   ready
DRAM:  256 MiB
NAND:  512 MiB
MMC:   OMAP SD/MMC: 0
In:    serial
Out:   serial
Err:   serial
Die ID #4d360001000000000160ae2c0600d00a
Net:   DaVinci-EMAC
Hit any key to stop autoboot:  0  ←入力  2秒以内にキー入力を行います
=>
```

- ⑦ microSD カードを使用できるように準備します。

```
=> mmc rescan 0 ←入力
```

- ⑧ microSD カード内の X-Loader を RAM にロードしてから、FlashROM に書き込みます。

```
=> fatload mmc 0 80000000 MLO ←入力
reading MLO

17072 bytes read
=> nandecch hw ←入力
HW ECC selected
=> nand erase 0 00020000 ←入力

NAND erase: device 0 offset 0x0, size 0x20000
Erasing at 0x0 -- 100% complete.
OK
=> nand write 80000000 0 $filesize ←入力

NAND write: device 0 offset 0x0, size 0x42b0
17072 bytes written: OK
```


付録 A. 付属品について

• PC-USB-03

付属の PC-USB-03 は、弊社のインターフェースコンバータの PC シリーズで販売している製品と同等品になります。詳細に関しては、別途マニュアルでご確認ください。

• USB ケーブル

付属の USB ケーブルは、USB2.0 に対応した AB タイプのケーブルとなります。本ケーブルは、PC-USB-03 とホスト PC を接続する時に使用します。

• JTAG-CNV-01

付属の JTAG-CNV-01 は、XG シリーズのオプション品として販売している製品と同等になります。詳細に関しては、別途マニュアルでご確認ください。

• microSD

付属の microSD カードは、Transcend 社製の 4GByte の microSDHC カード(Class 4)となります。弊社出荷時には、Android 等のデータは何も入っておりません。

• LAN ストレートケーブル

付属の LAN ストレートケーブルは、エンハンスドカテゴリ 5 (1000BASE-T)に対応した LAN ケーブルで、RoHS 指令に対応しています。

• AC アダプタ

付属の AC アダプタは、RoHS 指令に対応し、PSE 認定を取得しています。AC アダプタの仕様は、以下になります。

項目	仕様
入力	AC 100-240V 50/60Hz
出力	DC 5.0V 2.0A

Table A-1 AC アダプタ仕様

※付属品の内容及び仕様は予告なしに変更されることがありますのでご了承ください。

参考文献

VMware Player については以下の URL を参考にしてください。

- ・ VMware 社ホームページ
<http://www.vmware.com/jp/>
- ・ VMware Player 製品ホームページ
<http://www.vmware.com/jp/products/player/>

謝辞

Android、Linux、U-Boot の開発に関わった多くの貢献者に深い敬意と感謝の意を示します。

著作権について

- ・ 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。

商標について

- ・ AM3517 は、TEXISAS INSTRUMENTS 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。
- ・ 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®7、Windows®Vista、Windows®XP は、米国 Microsoft Corporation.の商品名称です。
- ・ VMware、VMware Player は、米国 VMware Inc.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®7 は、Windows 7 もしくは Win7
Windows®Vista は、Windows Vista もしくは WinVista
Windows®XP は、Windows XP もしくは WinXP
- ・ 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。



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