

XG Series

LK-3352-A02

Cortex-A8 AM3352 CPU BOARD

Install Manual

Rev 1.1

ダイジェスト版



ALPHA PROJECT
<https://www.apnet.co.jp>

目次

1. 概要	1
1.1 はじめに	1
1.2 Linuxについて	1
1.3 VirtualBoxについて	1
1.4 Ubuntuについて	1
1.5 GNUとFSFについて	2
1.6 GPLとLGPLについて	2
1.7 保証とサポート.....	2
2. 開発環境	3
2.1 ハードウェアについて	3
2.2 ソフトウェアについて	3
3. VirtualBox	4
3.1 VirtualBoxについて	4
3.2 インストール手順	5
3.3 仮想マシンの作成手順	9
4. Ubuntu	13
4.1 インストール手順	13
4.2 端末の起動方法.....	23
4.3 ネットワーク設定.....	25
4.4 NFS	29
4.5 TFTP.....	31
4.6 FTP.....	33
5. クロス開発環境	34
5.1 クロス開発環境について	34
5.2 SDKインストール (カーネルビルド用).....	35
5.3 toolchainインストール (rootfsビルド用)	40
5.4 パッケージのインストール	41
6. 製品サポートのご案内	43
7. エンジニアリングサービスのご案内	44

1. 概要

1.1 はじめに

XGシリーズは、TEXAS INSTRUMENTS社製Armマイクロプロセッサを搭載した組み込み用ボードコンピュータで、OSにLinuxを採用しています。

本製品では、開発環境にVirtualBoxを使用していますので、Windows上でのソフトウェア開発を行うことが可能です。

本ドキュメントでは、VirtualBoxのインストールからクロスコンパイラのインストールまでの開発環境の構築方法を説明します。

1.2 Linuxについて

Linuxとは1991年にLinus Torvalds氏によって開発された、オープンソースのUNIX互換オペレーティングシステムです。

Linuxはオープンソース、ロイヤリティフリーという特性から、世界中のプログラマたちにより日々改良され、今では大手企業のサーバや、行政機関などにも広く採用されています。

また、Linuxの特長としてCPUアーキテクチャに依存しないということがあげられます。これは、GNU Cコンパイラの恩恵にもよるものですが、数多くのターゲット(CPU)に移植されており、デジタル家電製品を中心に非PC系製品にも採用されるようになりました。

Linuxは、カーネルと呼ばれるOSの核となる部分とコマンドやユーティリティなど多くのソフトウェアから構成されます。

これらのソフトウェアの多くはFSFのGNUプロジェクトによるフリーソフトウェアです。

Linuxの詳細については、一般書籍やインターネットから多くの情報を得られますので、それらを参考にしてください。

1.3 VirtualBoxについて

VirtualBoxは現在オラクルによって提供されている仮想マシン実行ソフトウェアです。

VirtualBoxはWindows/Linux上で動作するPC/AT互換機エミュレータのため、これを使用することによりWindows上でLinuxを動作させることが可能となります。

Ver4.xからVirtualBox Open Source Edition(オープンソース版と呼ばれる)が公開され、そのライセンスは、GNU General Public License (GPL) Version2として配布されています。

1.4 Ubuntuについて

Ubuntuは、Ubuntuコミュニティにより開発されているオペレーティングシステムです。無償で提供されているため、ライセンス料を支払うことなくビジネス等で利用することができます。

長期サポート(LTS)版では5年間セキュリティアップデートが提供されます。

本開発環境では、VirtualBoxで作成した仮想環境に、このUbuntuをインストールして、そのOS環境で開発します。

2. 開発環境

VirtualBoxをインストールするPCが必要になります。

以下に、必要となるPCの環境を記載しますので、ご用意ください。

2.1 ハードウェアについて

本開発環境を利用するには、以下のハードウェア環境が必要になります。

なお、HDD空き容量やメモリは、最低限必要な容量となります。そのため、余裕がある場合には、それ以上の空き容量を確保していただくことを推奨します。

種類	環境
PC	PC/AT互換機
HDD空き容量	150GByte以上（Windowsが使用する容量は除きます。）
メモリ	8GByte以上
LANポート	1ポート（外部のネットワークに接続できる環境が必要です。）
CDドライブ もしくは DVDドライブ	開発キット付属のメディアが読み込み可能なドライブ （開発キットに付属するメディアによって必要なドライブが変わります。）

Table 2.1-1 ハードウェア環境



メモリを8GByte以上使用しますので、32bit版ではビルドできません。

Ubuntuに割り当てたメモリが8GByteの場合、ビルド時に仮想記憶によるスラッシングが発生するため、ビルド時間は非常に長くなります。

2.2 ソフトウェアについて

本開発環境を利用するには、以下のソフトウェア環境が必要になります。

種類	環境
ホストOS	Windows 10/11 64bit版
ゲストOS	Ubuntu 16.04LTS (16.04.4) 64bit版
VirtualBox	5.2.16以降
TI提供のSDK	PROCESSOR-SDK-LINUX-AM335X 04_00_00_04

Table 2.2-1 ソフトウェア環境



VirtualBoxのゲストOSに64bit版のOSを使用する場合には、CPUに以下のサポートが必要になります。

- ・インテル製CPUの場合は、仮想化支援機能「VT-x」のサポートが必要です。
- ・AMD製CPUの場合は、ロングモードでのセグメントリミットのサポートが必要です。

上記対応にはBIOSの設定が必要な場合があります。なお、BIOSの設定に関して、お使いのPCの説明書等でご確認ください。

3. VirtualBox

VirtualBoxは、仮想マシンを実行するソフトウェアです。VirtualBoxを利用することにより、Windows上でLinuxの開発を行うことができます。

本製品では、VirtualBoxのゲストOSにUbuntuを利用しています。

3.1 VirtualBoxについて

VirtualBoxは、Windows上でLinux環境を構築した仮想マシンを実行できるソフトウェアです。CPUを全てエミュレートするのではなく、カーネルモード命令のみをエミュレートすることでコマンド変換におけるオーバーヘッドを少なくし、実ハードウェアに近い性能を実現しています。またネットワーク機能も実ハードウェアと同等程度の速度で動作します。

本ドキュメントでは、VirtualBoxが動作するWindowsOSをホストOS、VirtualBox上で動作するLinuxOS (Ubuntu) をゲストOSと表現します。

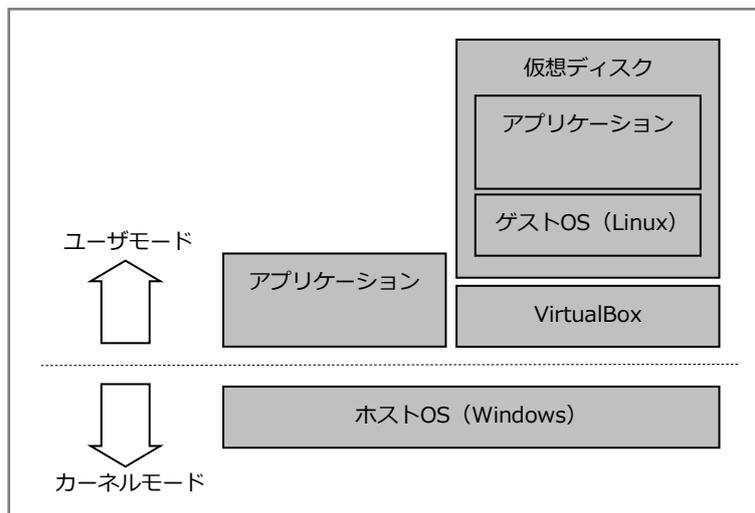


Fig 3.1-1 VirtualBox構成例

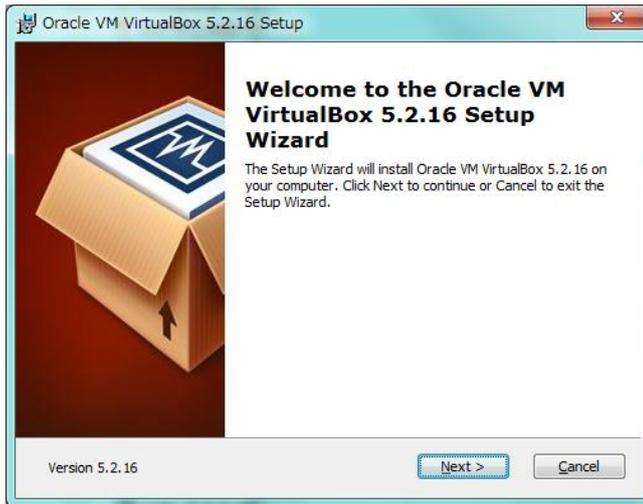
3.2 インストール手順

VirtualBoxのインストール手順を説明します。

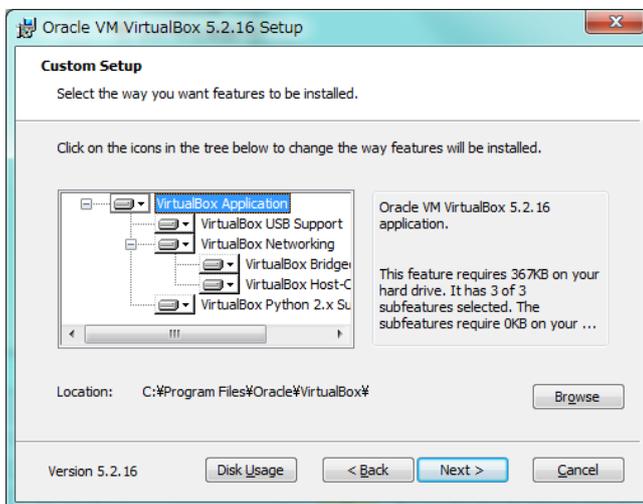
VirtualBoxのインストーラは本開発環境には付属していないため、VirtualBoxのダウンロードサイトから取得する必要があります。
なお、本ドキュメントで説明するバージョンに関しては、『2.2 ソフトウェアについて』でご確認ください。

- ① VirtualBoxのインストールを実行します。

インストーラ開始画面が表示されますので、『Next >』ボタンを押して開始します。



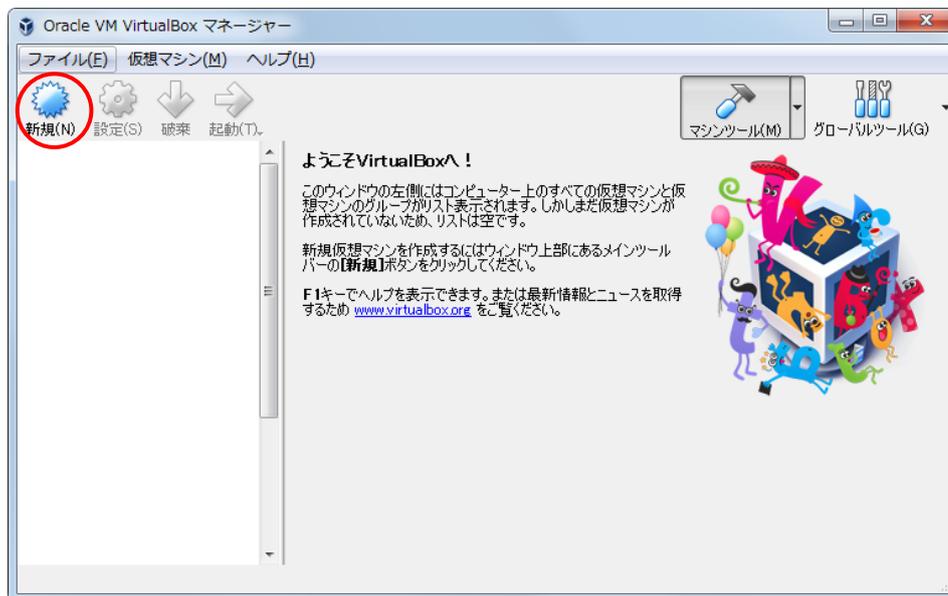
- ② インストール先フォルダ等を設定し、『Next >』ボタンを押します。



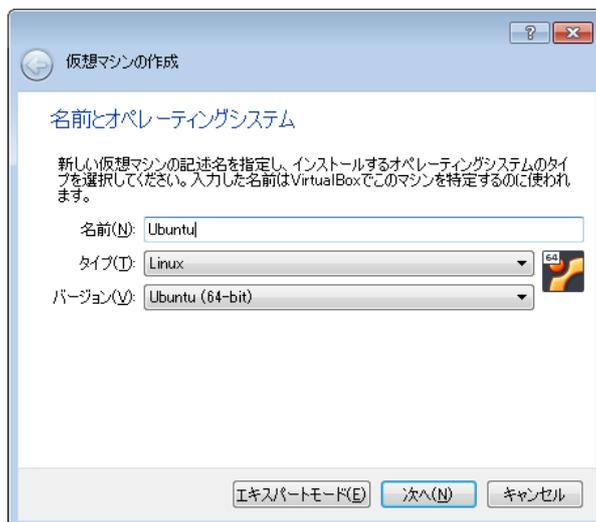
3.3 仮想マシンの作成手順

Ubuntuを動作させるため、仮想マシンを作成する必要があります。
作成するには、以下の手順で行います。

- ① VirtualBoxを起動して、左上の『新規(N)』ボタンを押します。



- ② 名前、タイプ等を入力して『次へ(N)』ボタンを押します。
名前は任意の名前(ここではUbuntuと入力することとします)、タイプはLinux、バージョンはUbuntu (64-bit)を選択します。



4. Ubuntu

本章では、Ubuntuのインストール手順を説明します。

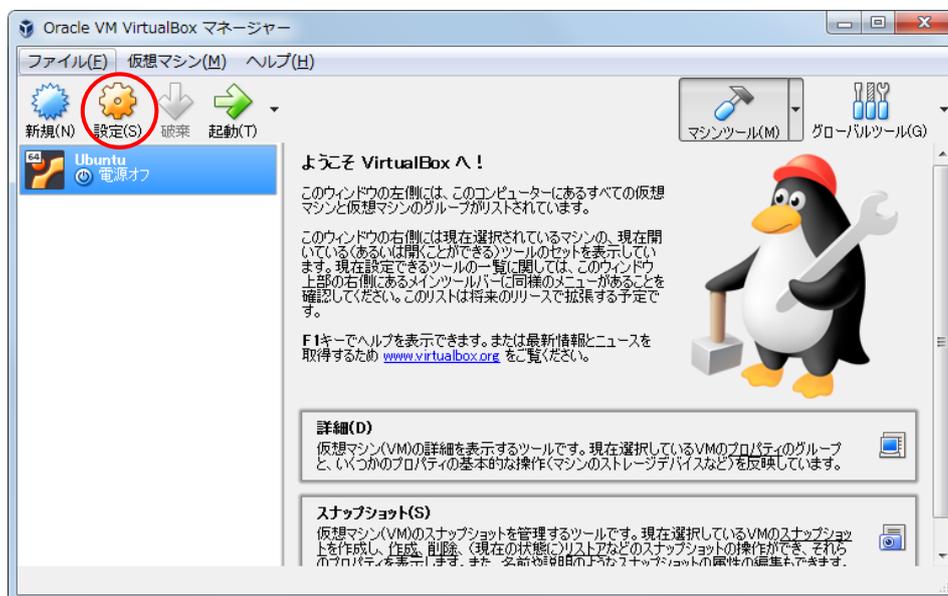
4.1 インストール手順

VirtualBoxで作成した仮想ディスクにUbuntuをインストールする手順を説明します。

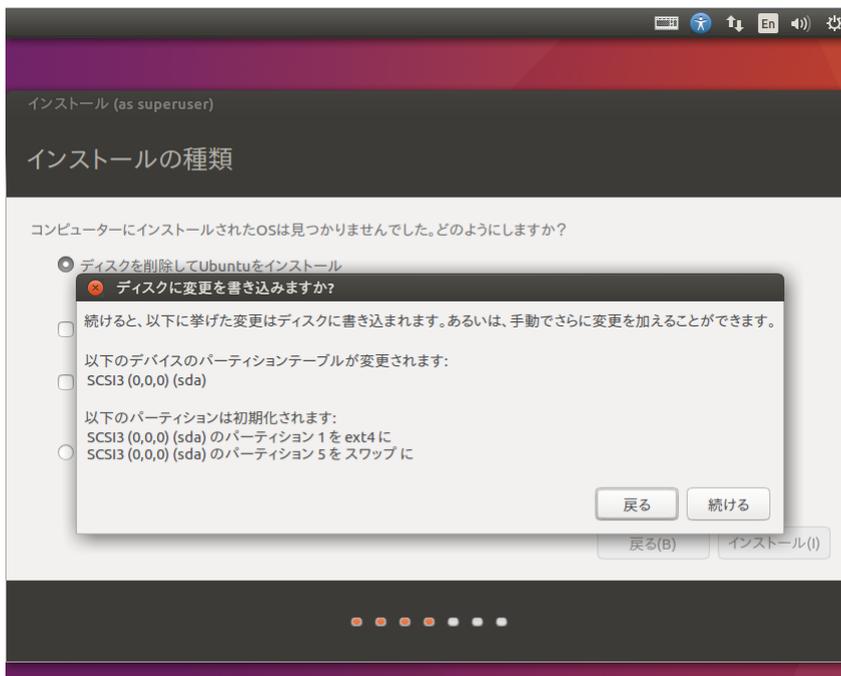
インストールにはUbuntuのISOイメージファイルが必要ですが、本開発環境には付属していないため、Ubuntuのダウンロードサイトから取得する必要があります。

なお、本ドキュメントで説明するバージョンに関しては、『2.2 ソフトウェアについて』でご確認ください。

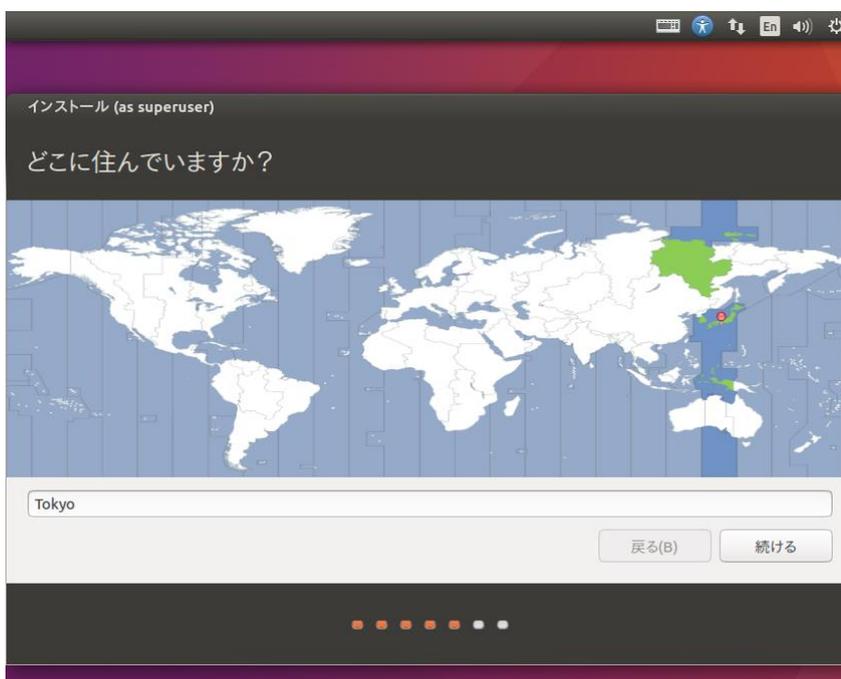
- ① VirtualBoxを起動し、左側からリスト『Ubuntu』を選択し、上部の『設定(S)』ボタンを押します。



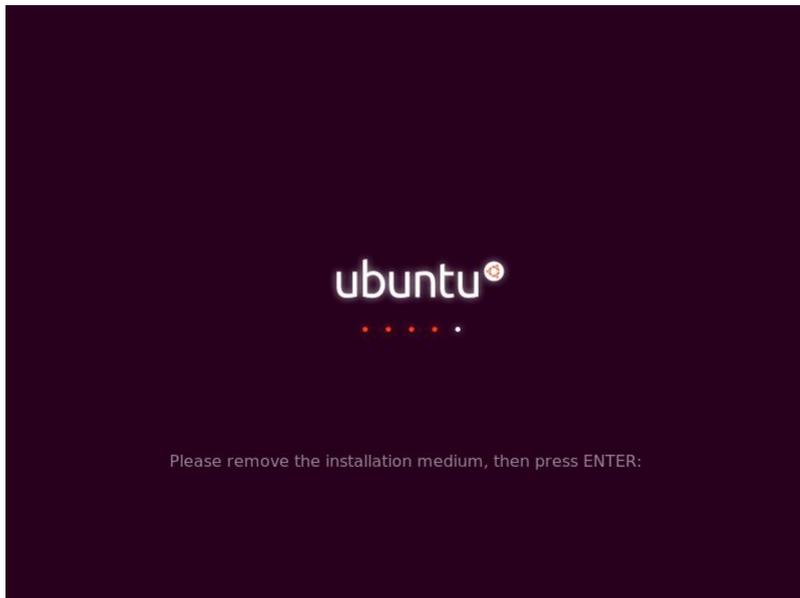
- ⑨ 削除の再確認の問い合わせがあります。『続ける』ボタンを押します。



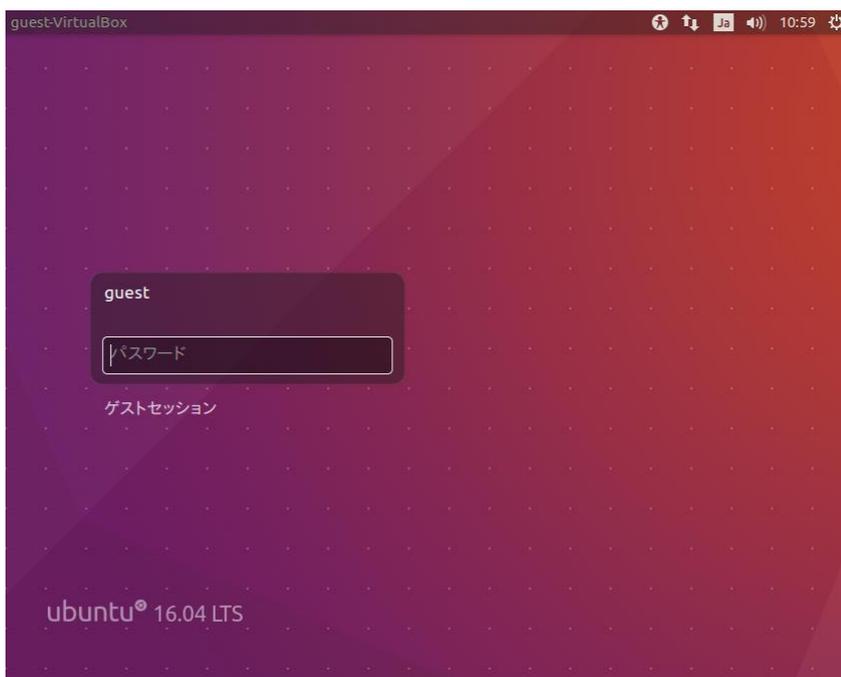
- ⑩ 地域を選択して『続ける』ボタンを押します。
本マニュアルでは、『Tokyo』を選択したとして手順を説明します。



- ⑮ 『Please remove installation medium, then press ENTER:』が表示されたらEnterキーを入力することでリセットされます。



- ⑯ 再起動後にログイン画面が表示されれば完了です。

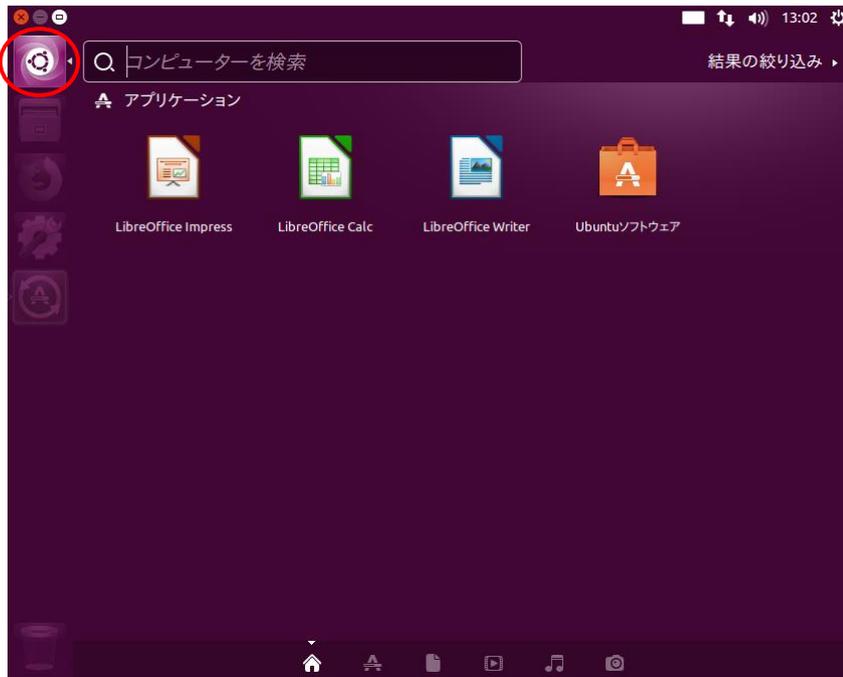


4.2 端末の起動方法

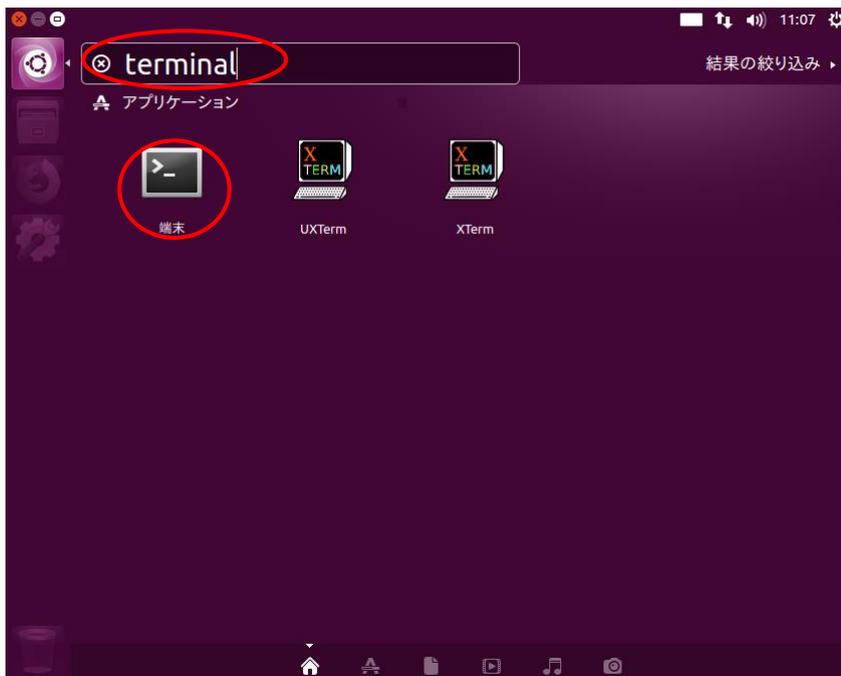
ゲストOSで操作を行う時には、コマンドラインにて操作を行うことがあります。

コマンドラインでの操作を行うには、端末アプリケーションを起動する必要があります。端末アプリケーションは、次の手順にて起動することができます。

- ① 左上の『Dashホーム』のアイコンをクリックします。



- ② 表示された検索入力に『terminal』と入力し、検索されたアプリケーションの『端末』をクリックします。



4.3 ネットワーク設定

ゲストOSのネットワーク設定を変更する方法について説明します。ゲストOSのネットワーク設定を変更する場合には、ネットワーク設定ファイルを書き換える必要があります。

以降の説明では、以下の設定値に変更することとします。

ネットワークの設定	
IPアドレス	192.168.128.210
サブネットマスク	255.255.255.0
ゲートウェイ	192.168.128.254
DNSサーバ	192.168.128.1

Table 4.3-1 ネットワーク設定



以降で説明する操作では、外部Webサイトにアクセスが必要な操作があります。

そのため、お客様の環境に応じて外部Webサイトにもアクセスできるように設定にさせていただきますようお願いいたします。

以降の説明では、上記に設定されていることを元に記述していますので、変更した場合にはその箇所は読み替えてお読みください。

- ① ネットワークボタンを押し、ポップアップメニューにて『接続を編集する...』を選択します。



4.4 NFS

NFS(Network File System)とはネットワークを介した分散ファイルシステムです。NFSを使用すれば、ゲストOS (Ubuntu) の共有ディレクトリのファイルを他のLinuxマシンから共有することができます。本章では、ゲストOS上でNFSサーバを起動し、共有ディレクトリにマウントします。

- ① 端末が起動していない場合は、起動します。
起動方法に関しては、『4.2 端末の起動方法』でご確認ください。
- ② 『apt』コマンドを使用して、インストールを行います。

```
$ sudo apt install nfs-kernel-server
パッケージリストを読み込んでいます... 完了
依存関係ツリーを作成しています
状態情報を読み取っています... 完了
以下の追加パッケージがインストールされます:
  keyutils libnfsidmap2 libtirpc1 nfs-common rpcbind
提案パッケージ:
  open-iscsi watchdog
以下のパッケージが新たにインストールされます:
  keyutils libnfsidmap2 libtirpc1 nfs-common nfs-kernel-server rpcbind
アップグレード: 0 個、新規インストール: 6 個、削除: 0 個、保留: 95 個。
467 kB のアーカイブを取得する必要があります。
この操作後に追加で 1,874 kB のディスク容量が消費されます。
続行しますか [Y/n]? Y
取得:1 http://jp.archive.ubuntu.com/ubuntu/ precise/main libgssglue1 i386 0.3-4 [21.9 kB]
:
途中省略
:
systemd (229-4ubuntu21.1) のトリガを処理しています ...
ureadahead (0.100.0-19) のトリガを処理しています ...
```

- ③ NFS共有ディレクトリ『/nfs』を作成し、パーミッションを設定します。

```
$ sudo mkdir /nfs
$ sudo chmod 777 /nfs
```

5. クロス開発環境

XGシリーズのCPUボードで動作するLinuxカーネルやアプリケーションプログラムを作成するには、クロス開発環境を構築する必要があります。本章では、XGシリーズ用のクロス開発環境をインストールする手順を説明します。

5.1 クロス開発環境について

クロス開発環境とは、以下の図のように実際に動作するシステム(CPUボード)とは異なるシステム(PC/AT互換機)で開発ができる環境のことを言います。

クロス開発環境で使用する特殊なコンパイラやアセンブラをクロスコンパイラやクロスアセンブラと呼びます。

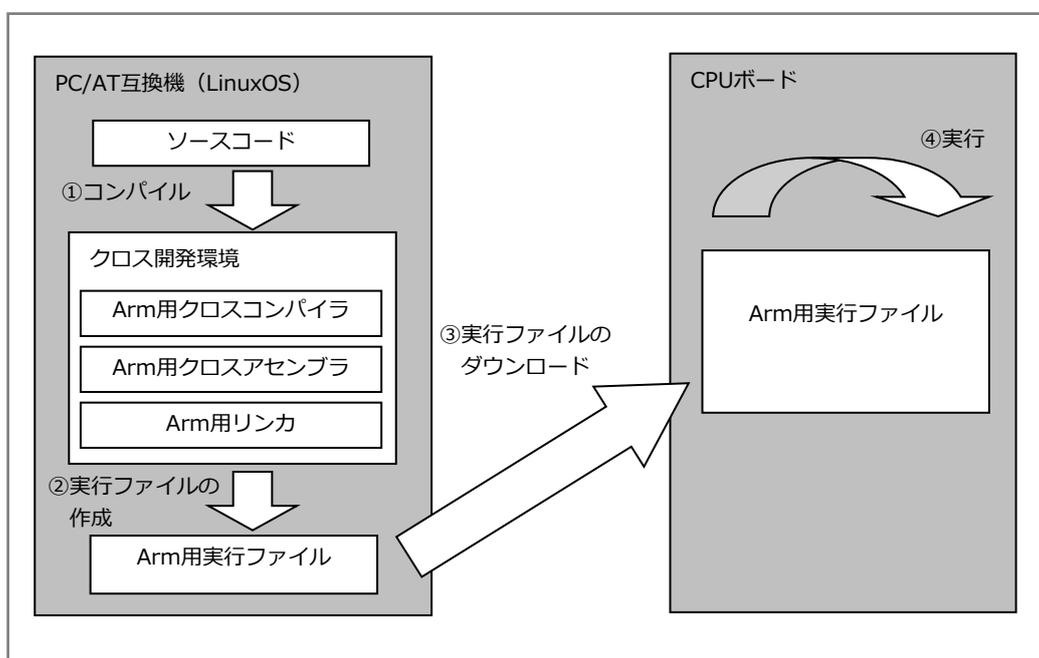


Fig 5.1-1 クロス開発環境

5.2 SDKインストール (カーネルビルド用)

TI提供のSDKのインストール手順を説明します。

SDKのインストーラは本開発環境には付属していないため、TIのダウンロードサイトから取得する必要があります。

なお、本ドキュメントで説明するバージョンに関しては、『2.2 ソフトウェアについて』でご確認ください。

また、ダウンロードした場所は、guestユーザのホームディレクトリ『/home/guest』に保存したとして説明します。

- ① 端末が起動していない場合は、起動します。
起動方法に関しては、『4.2 端末の起動方法』でご確認ください。
- ② インストールファイルをダウンロードしたホームディレクトリに移動します。

```
$ cd
```

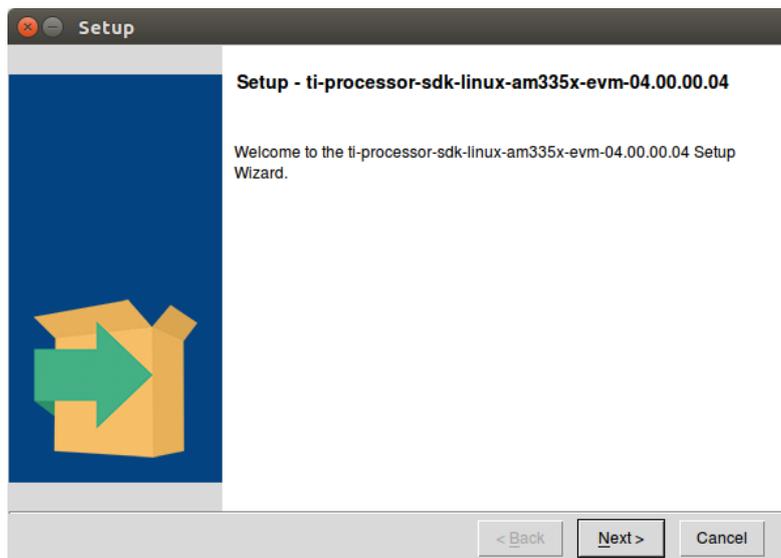
- ③ インストール用のファイルに実行権限を設定します。

```
$ chmod a+x ti-processor-sdk-linux-am335x-evm-04.00.00.04-Linux-x86-Install.bin
```

- ④ インストーラを実行します。

```
$ ./ti-processor-sdk-linux-am335x-evm-04.00.00.04-Linux-x86-Install.bin
```

- ⑤ 開始のメッセージです。
他のアプリケーションを終了させる等を行い『Next >』ボタンを押して先に進みます。



5.3 toolchainインストール (rootfsビルド用)

- ① 端末が起動していない場合は、起動します。
起動方法に関しては、『4.2 端末の起動方法』でご確認ください。

- ② ホームディレクトリに移動します。

```
$ cd
```

- ③ wgetコマンドでダウンロードします。

```
$ wget https://releases.linaro.org/components/toolchain/binaries/6.2-2016.11/arm-linux-gnueabi/gcc-linaro-6.2.1-2016.11-x86_64_arm-linux-gnueabi.tar.xz
```



上記では、表示上2行となっていますが、1行目の最後は改行をいれずに入力してください。

- ④ ファイルを解凍します。

```
$ tar -axf gcc-linaro-6.2.1-2016.11-x86_64_arm-linux-gnueabi.tar.xz
```

- ⑤ ホームディレクトリに解凍できたことを確認します。

```
$ ls ~/gcc-linaro-6.2.1-2016.11-x86_64_arm-linux-gnueabi
arm-linux-gnueabi  bin  gcc-linaro-6.2.1-2016.11-linux-manifest.txt  include  lib  libexec  c  share
```

5.4 パッケージのインストール

その他必要なライブラリ等のパッケージを『apt』コマンドを使用してインストールします。
インストールするパッケージは以下になります。

```
util-linux
device-tree-compiler
build-essential
dkms
linux-headers-generic
lzma
lzop
git
diffstat
texinfo
gawk
chrpath
libc6:i386
libstdc++6:i386
libz1:i386
libncurses5:i386
libncurses5-dev:amd64
```



上記の必要なパッケージに関しては、依存関係によりインストールされるパッケージは省略しております。

- ① 端末が起動していない場合は、起動します。
起動方法に関しては、『4.2 端末の起動方法』でご確認ください。

- ② パッケージ情報を更新します。

```
$ sudo apt update ←入力
```

- ③ 必要パッケージのインストールを行います。

```
$ sudo apt install util-linux ←入力
```

```
$ sudo apt install device-tree-compiler ←入力
```

```
$ sudo apt install build-essential dkms linux-headers-generic ←入力
```

```
$ sudo apt install lzma lzop ←入力
```

```
$ sudo apt install git diffstat texinfo gawk chrpath ←入力
```

```
$ sudo apt install libc6:i386 libstdc++6:i386 libz1:i386 ←入力
```

```
$ sudo apt install libncurses5:i386 libncurses5-dev:amd64 ←入力
```

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。が、万一不審な点、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

- ・VirtualBoxは、Oracle Corporation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。
- ・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®11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®11は、Windows 11もしくはWin11
- ・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
静岡県浜松市中央区積志町834
<https://www.apnet.co.jp>
E-Mail : query@apnet.co.jp
