

Intel<sup>®</sup> SoC Board Series

Linux 開発

ARM CPU BOARD

# Install Manual for SA-CY500S

Rev 1.4

ダイジェスト版



**AP ALPHA PROJECT**

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

©2023 Alpha Project Co., Ltd.

## 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 Ubuntu について.....	13
4.2 インストール手順.....	13
4.3 端末の起動方法.....	29
4.4 ネットワーク設定.....	31
4.5 レポジトリの変更.....	34
4.6 NFS.....	35
4.7 TFTP.....	37
4.8 FTP.....	39
5. クロス開発環境	41
5.1 クロス開発環境について.....	41
5.2 SDK インストール手順.....	42

5.3	パッケージのインストール手順 .....	47
5.4	dash 処理の設定.....	49
5.5	Toolchain インストール手順 .....	50
6.	製品サポートのご案内	51
7.	エンジニアリングサービスのご案内	52

# 1. 概要

## 1.1 はじめに

SA-Cy シリーズは、インテル社製 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ーや、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で 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(ウブントゥ)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。

デスクトップについて 3 年間、サーバーについては 5 年間の提供期間がある長期サポート(LTS)版も用意されています。

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	20GByte 以上 (Windows が使用する容量は除きます。)
メモリ	2GByte 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

### 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 12.04LTS (12.04.4) 64bit 版
VirtualBox	5.0.2 以降
Intel 提供の SDK	SoC Embedded Design Suite v15.0.1.60
Linaro 提供の Toolchain	gcc-linaro-arm-linux-gnueabi-4.9-2014.09
開発環境	インテル®Quartus® Prime <sup>※</sup>

Table 2.2-1 ソフトウェア環境

- ※ Preloader, ブートローダの作成にはインテル®Quartus® Prime で作成した FPGA のプロジェクトが必要になります。  
詳細に関しては『AN5102 SA-Cy500S(Cyclone® V SoC Board) サンプル HDL プログラム解説』を参照してください。



64bitOS の仮想マシンを作成するためにはホスト PC の BIOS 設定が必要です。  
ホスト PC の BIOS 画面で仮想化支援機構(VT-x/AMD-V)を有効にしてください。  
BIOS の設定手法に関してはホスト PC のマニュアルを参照してください。



本手順では、VirtualBox のバージョン 5.0.2 にて説明しています。それ以降のバージョンでは、画面が異なる場合もございますので、適宜読みかえて行ってください。

## 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。

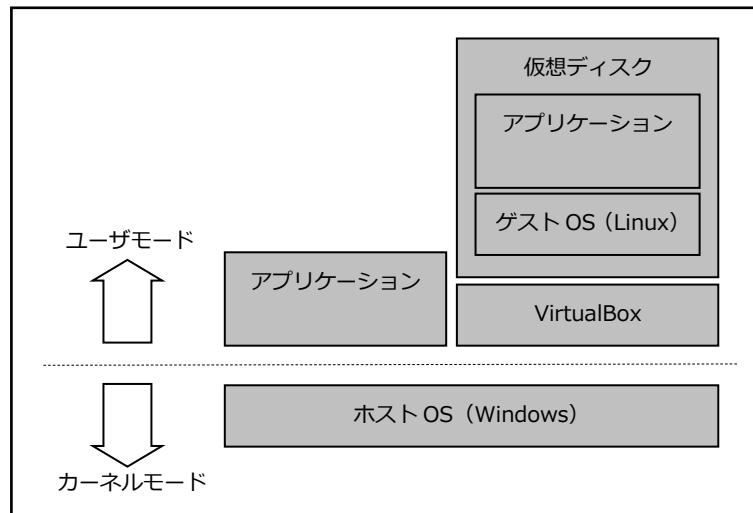


Fig 3.1-1 VirtualBox 構成例

## 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。

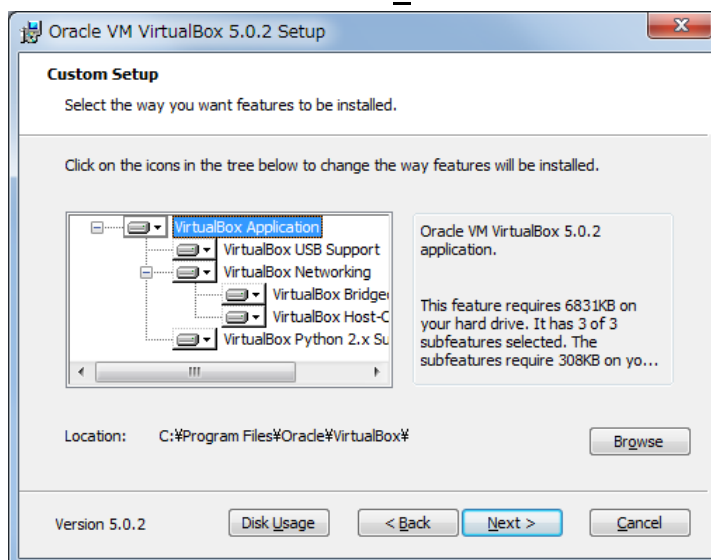
なお、本ドキュメントで説明するバージョンに関しては、『[2.2 ソフトウェアについて](#)』でご確認ください。

- 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。



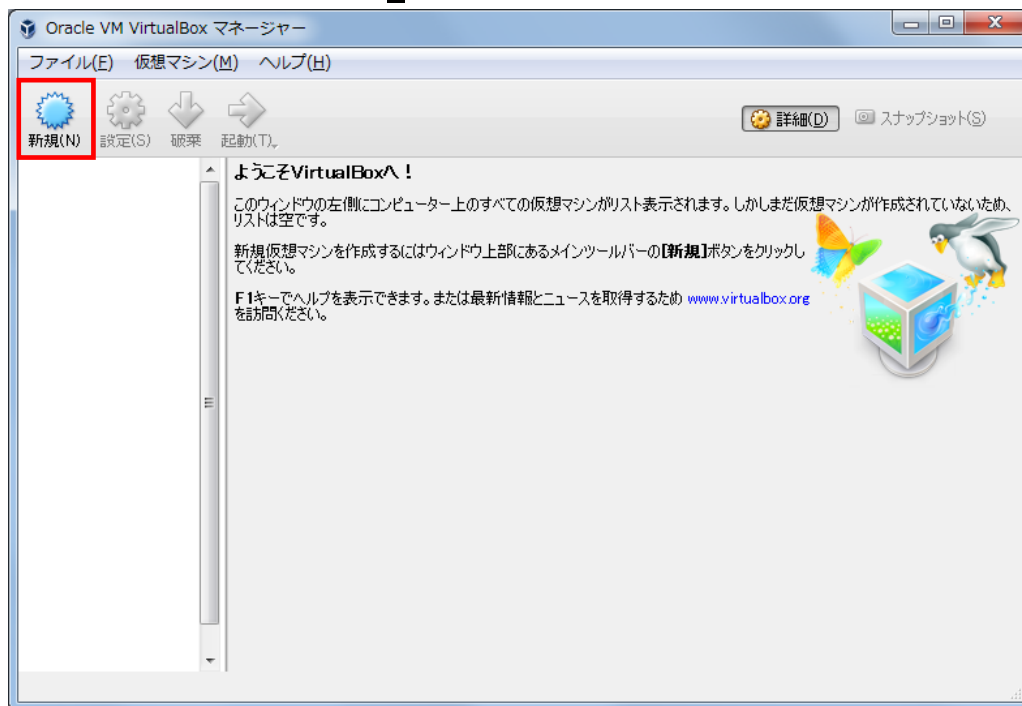
- ② インストール先フォルダ等を設定し、『**Next >**』ボタンを押します。



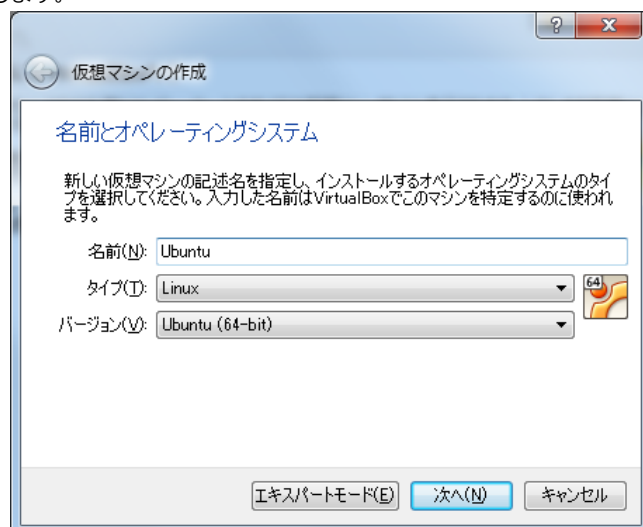
### 3.3 仮想マシンの作成手順

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(64bit)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 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。

デスクトップについて 3 年間、サーバーについては 5 年間の提供期間がある長期サポート(LTS)版も用意されています。

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。

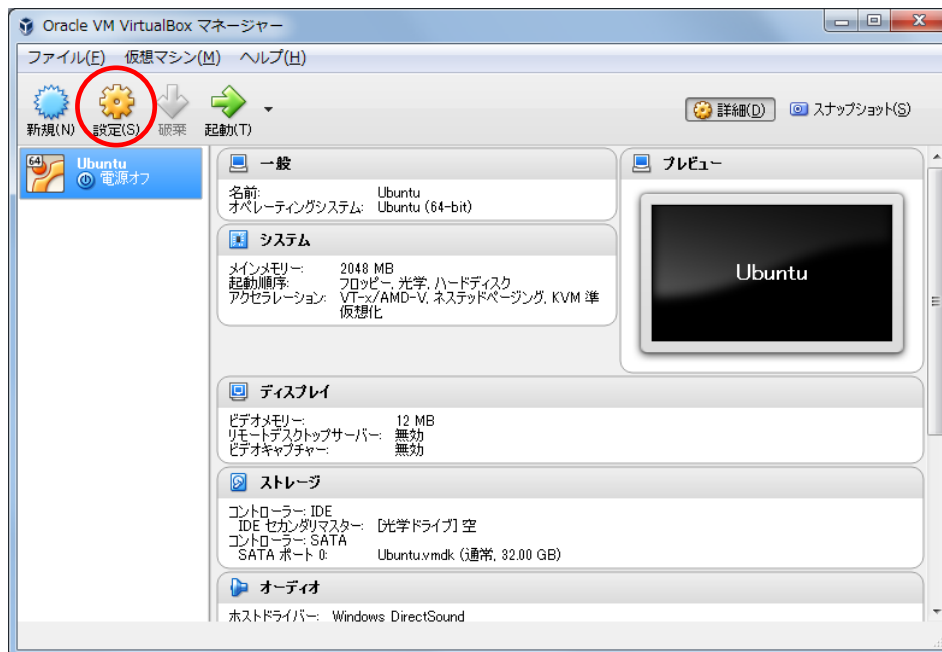
### 4.2 インストール手順

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。

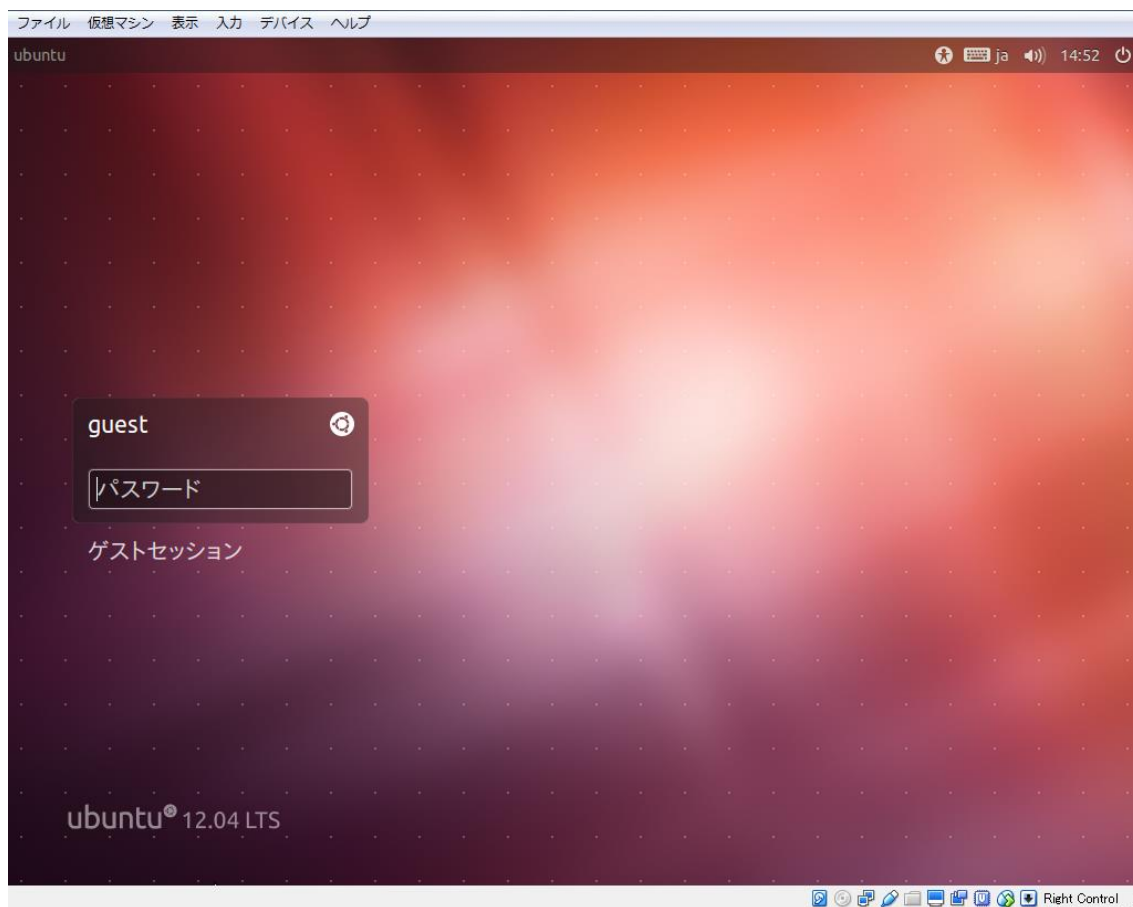
インストールには Ubuntu の ISO イメージファイルが必要ですが、本開発環境には付属していないため、Ubuntu のダウンロードサイトから取得する必要があります。

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。



- ⑱ 再起動後にログイン画面が表示されれば完了です。



## 4.3 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 下側に表示されているアイコンの左から 2 番目をクリックして、その後に、中央の『インストール済み』の『さらに 78 個の結果を表示』をクリックします。



## 4.4 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 4.4-1 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。



ネットワークの設定方法は、いくつかありますが、今回の方法は、network-manager を停止させて、interfaces ファイルを変更する方法で説明します。

- 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、『[4.3 端末の起動方法](#)』でご確認ください。

- ② network-manager を停止します。

```
省略 $ sudo service network-manager stop ←入力
[sudo] password for guest: ←入力
network-manager stop/waiting
```

- ③ network-manager を次回から起動しないようにします。

```
省略 $ sudo update-rc.d -f network-manager remove ←入力
[sudo] password for guest: ←入力
Removing any system startup links for /etc/init.d/network-manager ...
```

## 4.6 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、『[4.3 端末の起動方法](#)』でご確認ください。

- ② 『`apt-get`』コマンドを使用して、インストールを行います。

```

省略 $ sudo apt-get install nfs-kernel-server
[sudo] password for guest:
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
以下の特別パッケージがインストールされます:
  libgssglue1 libnfsidmap2 libtirpc1 nfs-common rpcbind
以下のパッケージが新たにインストールされます:
  libgssglue1 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: 376 個。
548 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 1,923 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-updates/main libgssglue1 amd64 0.3-
4ubuntu0.1 [22.5 kB]
:
途中省略
:
* Starting NFS kernel daemon
libc-bin のトリガを処理しています ...
ldconfig deferred processing now taking place
  
```

- 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。

```

省略 $ sudo mkdir /nfs
[sudo] password for guest:
省略 $ sudo chmod 777 /nfs
[sudo] password for guest:
  
```

## 5. クロス開発環境

SA-Cy シリーズの CPU ボードで動作するブートローダや Linux カーネル、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 SA-Cy 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。

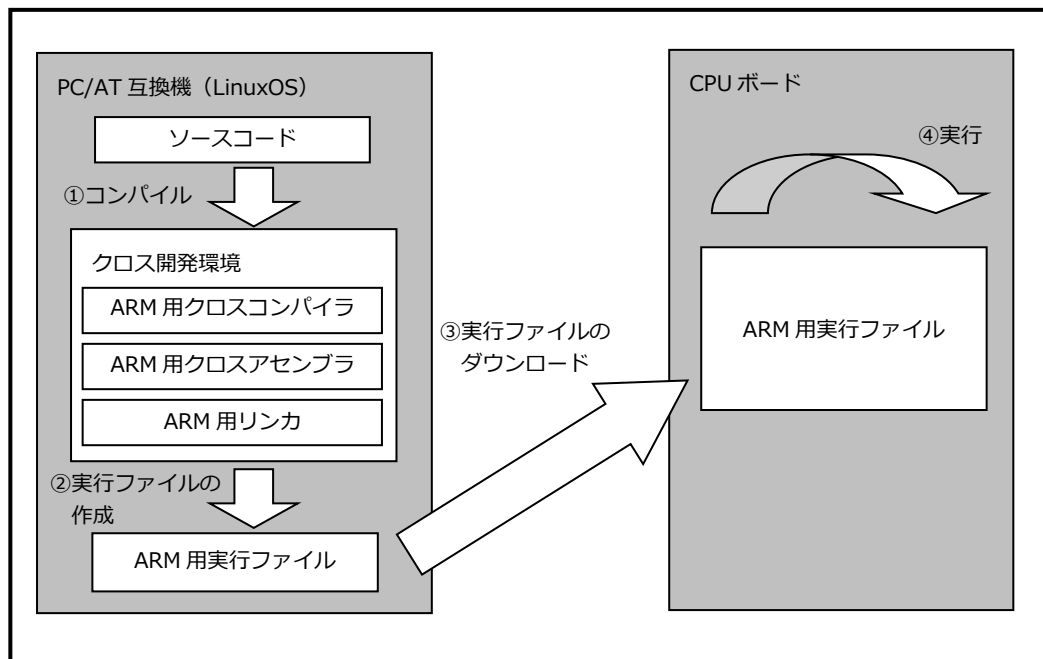


Fig 5.1-1 クロス開発環境

## 5.2 SDK インストール手順

インテル社提供の 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、インテル社のダウンロードサイト

(<https://fpgasoftware.intel.com/soceds/15.0.1/?edition=web&platform=linux>)から取得する必要があります。

その際、インテル社のサイトへのログインが要求されますので必要に応じてアカウントの作成およびログインを行ってください。

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](#)』をご確認ください。

また、ダウンロードしたインストーラは guest ユーザのホームディレクトリ『[/home/guest](#)』に保存されたとして説明します。



Ubuntu 12.04 で採用されているブラウザ (Firefox) ではインテル社の Web サイトにアクセスすることができません。

そのため、SDK のダウンロードはホスト PC にインストールされているブラウザで行い、ダウンロードしたインストーラ (SoCEDSSetup-15.0.1.60-linux.run) を『[4.8 FTP](#)』などを用いてゲスト OS にコピーしてください。

- 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、『[4.3 端末の起動方法](#)』をご確認ください。

- 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。

```
省略 $ 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。

```
省略 $ chmod a+x SoCEDSSetup-15.0.1.60-linux.run
```

- ④ インストーラを実行します。

```
省略 $ sudo ./SoCEDSSetup-15.0.1.60-linux.run
```

- ⑤ 開始の確認メッセージが表示されます。  
問題なければ、『Next >』ボタンを押して、インストールを開始します。

## 5.3 パッケージのインストール手順

その他必要なライブラリ等のパッケージを『**apt-get**』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。

```
sed
wget
cvs
subversion
git-core
coreutils
unzip
texi2html
texinfo
libsdl1.2-dev
docbook-utils
gawk
python-pysqlite2
diffstat
help2man
make
gcc
build-essential
g++
desktop-file-utils
chrpath
libgl1-mesa-dev
libglu1-mesa-dev
mercurial
autoconf
automake
groff
libtool
xterm
ia32-libs
uboot-mkimage
device-tree-compiler
```



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。



```

省略 $ sudo apt-get update; sudo apt-get upgrade; sudo apt-get install sed wget cvs subversion
git-core coreutils unzip texi2html texinfo libsdl1.2-dev docbook-utils gawk python-pysqlite2
diffstat help2man make gcc build-essential g++ desktop-file-utils chrpath libgl1-mesa-dev
libglu1-mesa-dev mercurial autoconf automake groff libtool xterm uboot-mkimage ia32-libs
device-tree-compiler ←入力
[sudo] password for guest: ←入力
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

:
途中省略
:

libqt4-declarative:i386 (4:4.8.1-0ubuntu4.9) を設定しています ...
libc-bin のトリガを処理しています ...
ldconfig deferred processing now taking place
libgdk-pixbuf2.0-0 のトリガを処理しています ...
libgdk-pixbuf2.0-0:i386 のトリガを処理しています ...

```



入力に際し、改行前後の単語は繋がっていません。

『subversiongit-core』ではなく、『subversion git-core』となります。  
2行目以降も同様です。

実行中に以下のようなメッセージが表示されたらYを入力してください。

```

この操作後に追加で 24.0 MB のディスク容量が消費されます。
続行しますか [Y/n]? Y ←入力

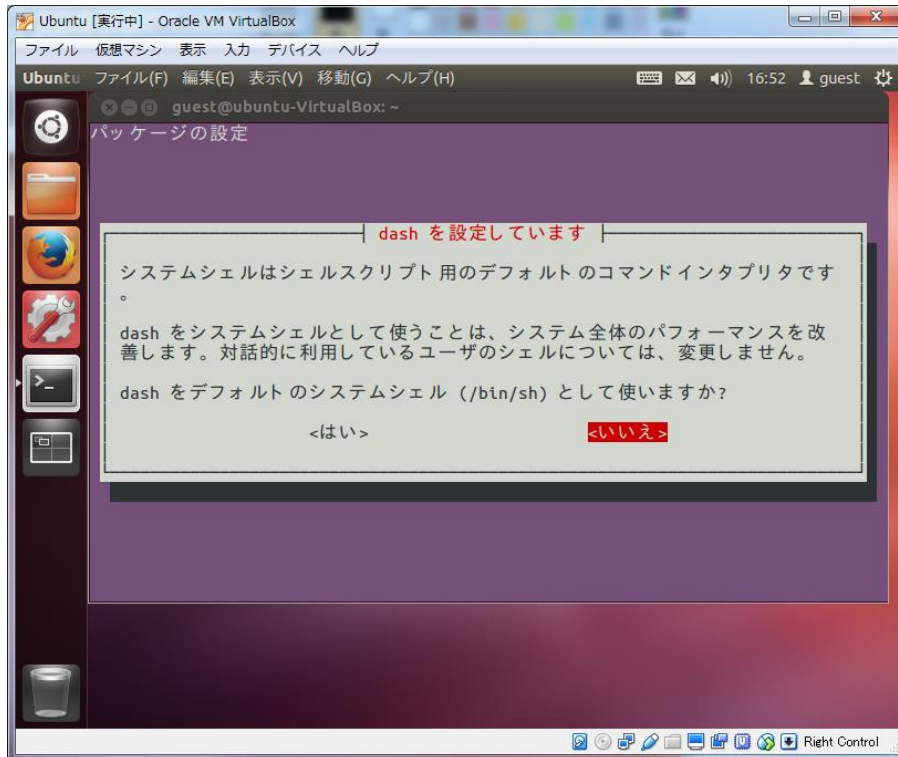
```

## 5.4 dash 処理の設定

dash の設定を変更します。

```
$ sudo dpkg-reconfigure dash
[sudo] password for guest:
```

入力後、以下の画面が表示されますのでいいえを選択し、エンターを押します。



## 5.5 Toolchain インストール手順

Linaro 提供の Toolchain インストール手順を説明します。

Toolchain は本開発環境には付属していないため、Linaro 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 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、『[4.3 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。

```
省略 $ cd ←入力
```

- ③ Toolchain を保存するディレクトリ『**socfpga-toolchain**』を作成します。

```
省略 $ mkdir socfpga-toolchain ←入力
```

- ④ Linaro から Toolchain のダウンロードを行います。

```
省略 $ wget http://releases.linaro.org/archive/14.09/components/toolchain/binaries/gcc-linaro-arm-linux-gnueabi-4.9-2014.09_linux.tar.xz ←入力

--2018-08-24 10:47:28--
http://releases.linaro.org/archive/14.09/components/toolchain/binaries/gcc-linaro-arm-linux-gnueabi-4.9-2014.09_linux.tar.xz
releases.linaro.org (releases.linaro.org) をDNSに問いあわせています... 13.228.101.204

:
途中省略
:

2018-08-24 10:47:38 (5.83 MB/s) - `gcc-linaro-arm-linux-gnueabi-4.9-2014.09_linux.tar.xz'
へ保存完了 [51278488/51278488]
```



入力に際し、改行前後の単語は連続した1つの単語となっています。  
『**gcc-linaro-arm-linux-gnueabi-4.9-2014.09\_linux.tar.xz**』となります。

- ⑤ ダウンロードした Toolchain を解凍し、『**socfpga-toolchain**』にリネームします。

```
省略 $ tar -xvf gcc-linaro-arm-linux-gnueabi-4.9-2014.09_linux.tar.xz --strip-components 1 -C socfpga-toolchain ←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。
- ・Cyclone® V および Cyclone® V SoC は Intel Corporation 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®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](mailto:query@apnet.co.jp)