

# E2 Spider セットアップ手引き書

2018/08/03 中村啓太, 遠橋健太, 原澤輝

## 使用したパソコン構成

- Intel NUC Core i7 搭載 小型 PC ベアボーン 2.5 インチ/M.2 SSD 対応 BOXNUC7I7BNH
- シリコンパワー ノート PC 用メモリ DDR4-2133(PC4-17000) 8GBX2 枚 260pin 1.2V CL15 永久保証 SP016GBSFU213B22
- WD 内蔵 SSD M.2-2280/512GB/WD Black/PCIe Gen3 NVMe/5 年保証/WDS512G1X0C

## 1 インテル® NUC キット NUC7i7BNH に Ubuntu16.04 をインストール

### 1.1 Ubuntu16.04 のインストール USB を作成

#### 1.1.1 事前準備

以下のものを用意する

- 初期化しても良い USB メモリ (容量は 2GB 程度で十分)
- <https://www.ubuntulinux.jp/News/ubuntu1604-ja-remix> から `ubuntu-ja-16.04-desktop-amd64.iso` をダウンロード

#### 1.1.2 Windows10 の場合

- 用意した USB メモリを差し込む。(どのドライブになるかを把握する)
- [https://ja.osdn.net/projects/sfnet\\_win32diskimager/releases/](https://ja.osdn.net/projects/sfnet_win32diskimager/releases/) から, `Win32DiskImager-1.0.0-binary.zip` をダウンロードする.
- `Win32DiskImager-1.0.0-binary.zip` を解凍して, `Win32DiskImager.exe` を起動する.
- `Win32DiskImager.exe` 画面 (図 1) において,
  - ① 「Image File のアイコン」をクリックし, ダウンロードした `ubuntu-ja-16.04-desktop-amd64.iso` を指定する.
  - ② 「Device」をクリックし, 差し込んだ USB メモリのドライブを指定する.
  - ③ 「Write」をクリックすることで, Ubuntu16.04 のインストール USB が作成される.

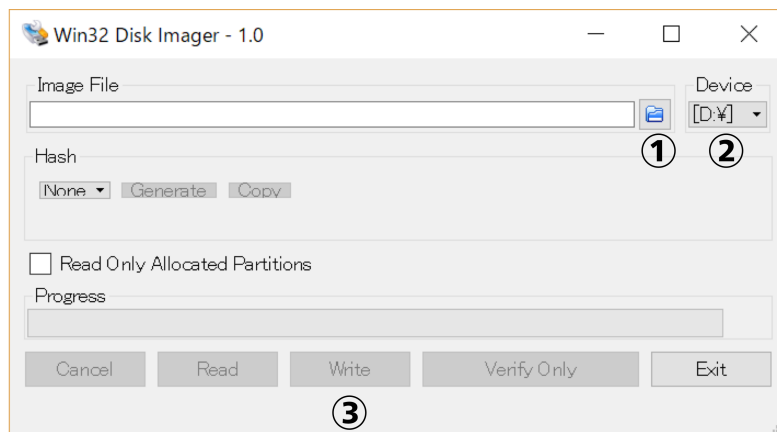


図 1 Win32DiskImager.exe 画面

### 1.1.3 macOS High Sierra の場合

- 用意した USB メモリを差し込む.
- Terminal.app を開く (/Applications/Utilities/Terminal.app)
- ダウンロードした ubuntu-ja-16.04-desktop-amd64.iso があるフォルダに移動する.

```
1 $ cd Downloads
```

これは例ですので、適宜変更してください.

- hdiutil コマンドで、ダウンロードした Ubuntu の iso ファイルを dmg ファイルに変換する.

```
1 $ hdiutil convert -format UDRW -o ./ubuntu-ja-16.04-desktop-amd64.img ./
  ubuntu-ja-16.04-desktop-amd64.iso
```

- ubuntu-ja-16.04-desktop-amd64.img.dmg が作成されるので、mv コマンドで dmg ファイルをを img ファイルとしてリネームする.

```
1 $ mv ./ubuntu-ja-16.04-desktop-amd64.img.dmg ./ubuntu-ja-16.04-desktop-
  amd64.img
```

- diskutil list コマンドで USB メモリのデバイスノードを確認する.

```
1 $ diskutil list
```

執筆者環境では、USB メモリのデバイスノードが『/dev/disk2』でしたので、『/dev/disk2』を使用してコマンドを記述します. 例として記述しますので、適宜変更してください.

- hdiutil コマンドで、USB メモリをアンマウントする.

```
1 $ diskutil unMountDisk /dev/disk2
```

- dd コマンドで、Ubuntu16.04 のインストール USB が作成される.

```
1 $ sudo dd if=./ubuntu-ja-16.04-desktop-amd64.img of=/dev/rdisk2 bs=1m
```

## 1.2 BIOS のファームウェアアップデート

### 1.2.1 事前準備

以下のものを用意する

- USB メモリ (FAT32 フォーマット推奨)

### 1.2.2 ファームウェアアップデート

- <https://downloadcenter.intel.com/ja/product/98414/-PC> にアクセスし、インテル® NUC キット NUC7i7BNH 用の BIOS アップデートページに移動して、\*\*\*\*\*.bio をダウンロードして、用意した USB メモリに保存する.
- \*\*\*\*\*.bio を保存した USB をインテル® NUC キットに差し込み、インテル® NUC キットの電源を入れて、「F2」キーを押して、BIOS 画面を立ち上げる。(図 2)
- 「F7」キーを押して、Update BIOS メニューを開く.
- 「Choose a .bio file」ダイアログが開くので、左側カラムにある差し込んだ USB に保存した\*\*\*\*\*.bio を選択し、「Open」をクリックする.

- 「Confirm BIOS Update」ダイアログが開くので、「Yes」をクリックし、アップデートを行う（しばらくするとアップデートが終了し、再起動する）
- 再起動終了後、差し込んだ USB メモリを抜く。

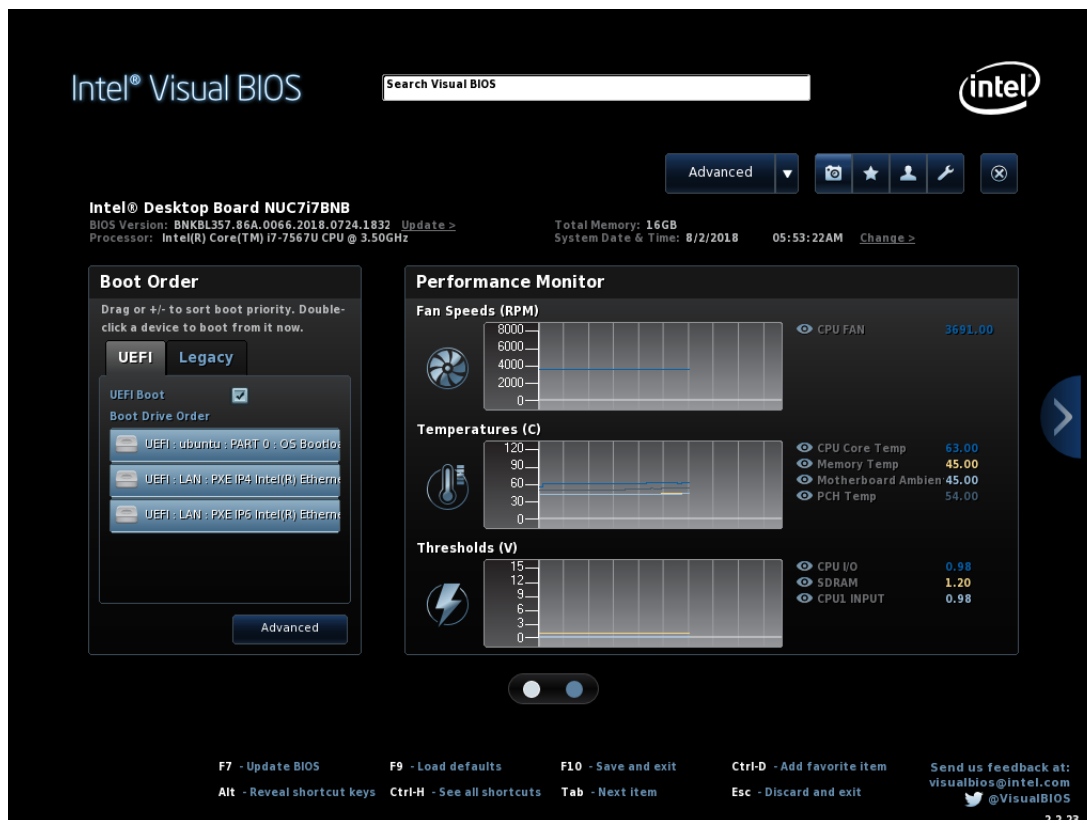


図 2 BIOS 画面

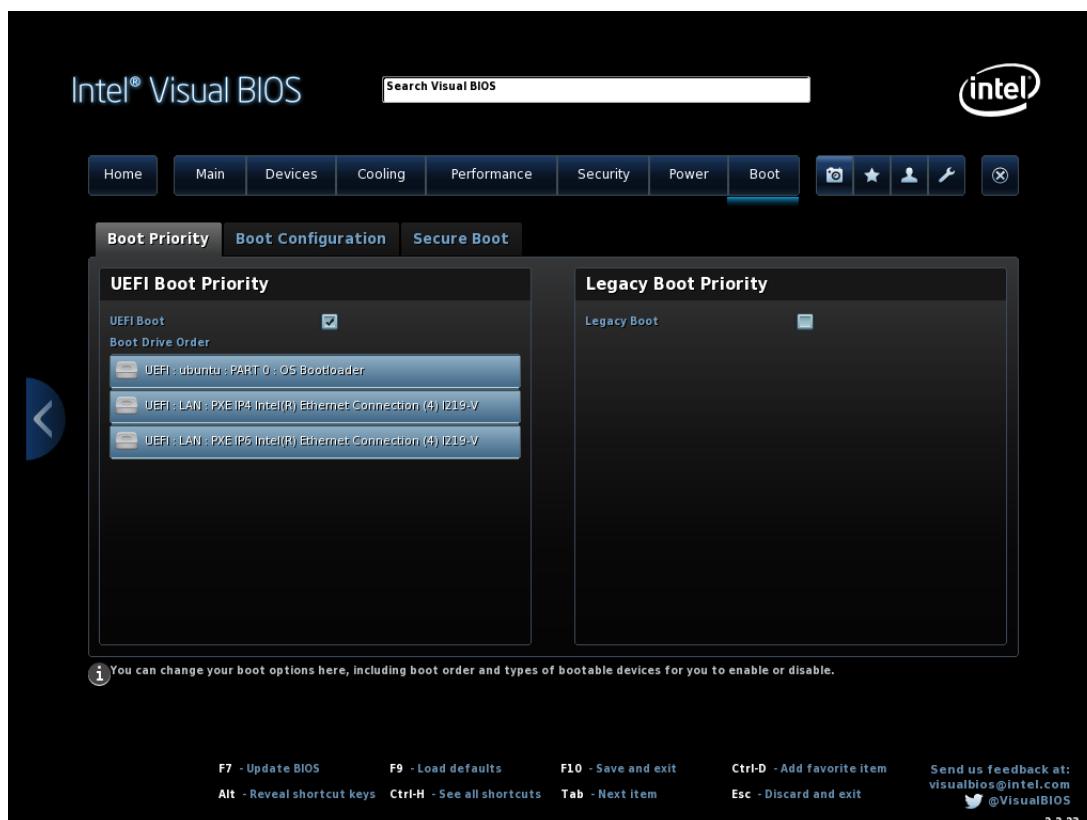


図 3 Legacy Boot Priority カラム画面

### 1.3 BIOS の設定

- インテル® NUC キットの電源を入れて、「F2」キーを押して、BIOS 画面を立ち上げる。（図 2）
- 「Advanced」をクリック、「Boot」をクリックし、右にある「Legacy Boot Priority」カラムの「Legacy Boot」

のチェックを外す。(図 3)

- 「F10」キーを押し、「Exit Visual BIOS」ダイアログが出るので、「Yes」をクリックして再起動する。

## 1.4 USB による Ubuntu16.04 のインストール

※写真は仮想環境下でインストールした場合のものを含んでいます。

- 作成した Ubuntu16.04 のインストール USB をインテル® NUC キットに指す。
- インテル® NUC キットの電源を入れて、「F2」キーを押して、BIOS 画面を立ち上げる。(図 2)
- 「F2」キーを押して、BIOS 画面を立ち上げる。
- 「Advanced」をクリックし、「Boot」をクリックし、左にある「UEFI Boot Priority」カラムにある Boot Drive Order の順番をドラッグアンドドロップで変更できるので、Ubuntu16.04 のインストール USB が最初に boot されるように変更する。(図 4)

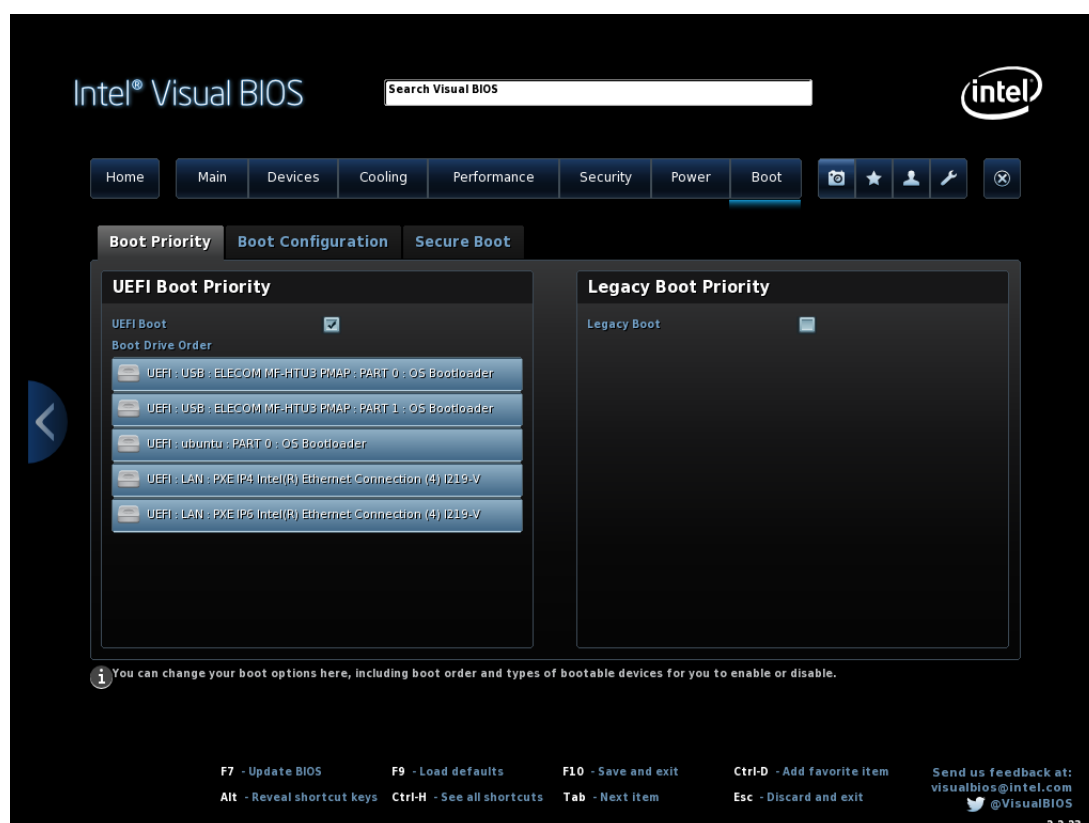


図 4 UEFI Boot Priority カラム画面

- 「F10」キーを押し、「Exit Visual BIOS」ダイアログが出るので、「Yes」をクリックして再起動する。
- 「Install Ubuntu」を選択し、Enter キーを押す(図 5)
- 「日本語」を選択し「続ける」をクリックする。(図 6)
- 「グラフィック・・・」のみを選択し「続ける」をクリックする。(図 7)
- 「ディスクを削除して・・・」を選択し「インストール」をクリックする。(図 8)
- ダイアログが出るので、「続ける」をクリックする。(図 9)
- 「Tokyo」と入力されていることを確認して、「続ける」をクリックする。(図 10)
- 左カラムに「日本語」、右カラムに「日本語」を選択して、「続ける」をクリックする。(図 11)
- 各情報を入力し、「ログイン時にパスワードを要求する」を選択し、「続ける」をクリックする。この手引き書では以下のように設定しています。(図 12)
  - あなたの名前: REL
  - コンピュータの名前: SPIDER-BOX
  - ユーザー名の入力: e2

- パスワードの名前：\*\*\*\*\*
- パスワードの確認：\*\*\*\*\*
- Ubuntu のインストールが始まる (図 13)
- インストール終了後、ダイアログが出るので「今すぐ再起動する」をクリックする (図 14)
- 「Please remove・・・」の画面が出たら、指してある USB メモ리를抜き、「Enter」キーを押す。(再起動しない場合は、「ctrl」キー、「alt」キー、「del」キーを同時に長押しすることで再起動する)

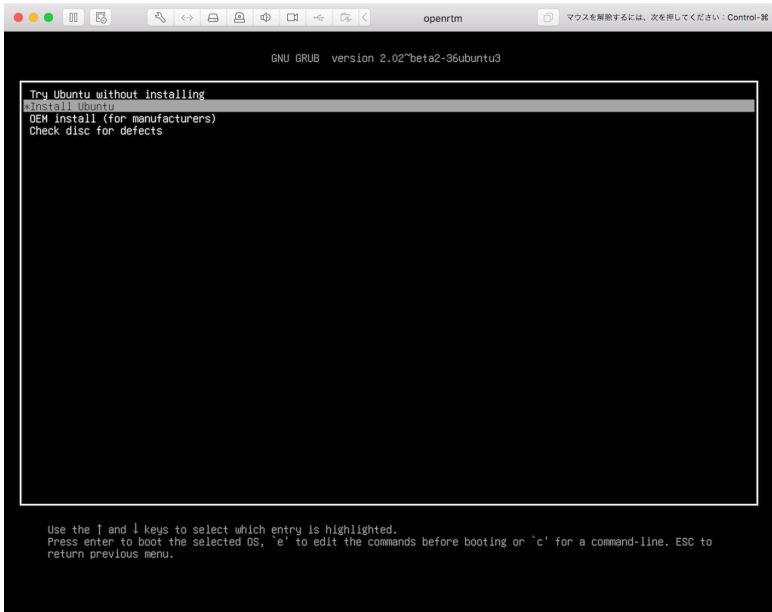


図 5 Ubuntu インストールメニュー画面

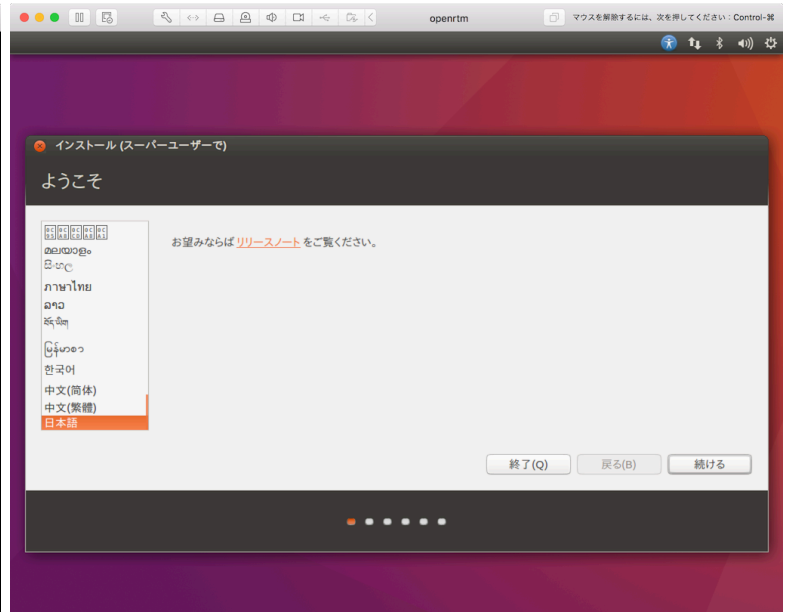


図 6 ようこそ

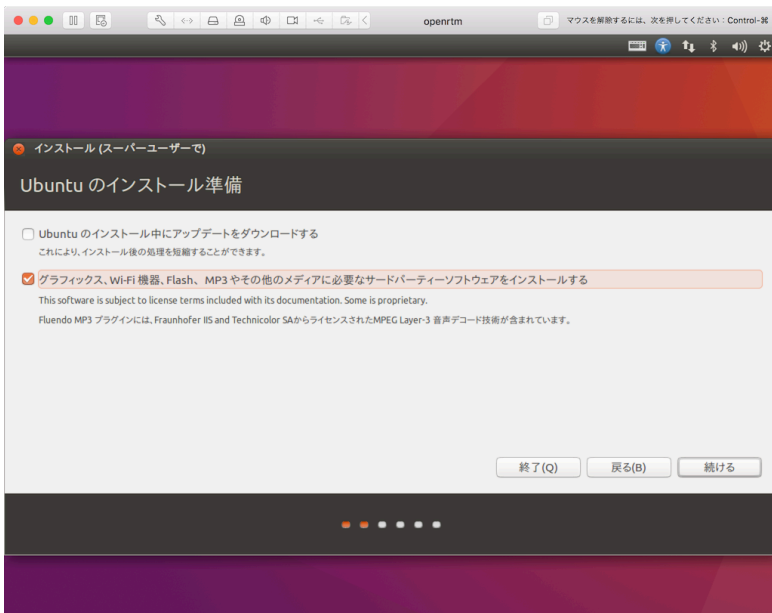


図 7 Ubuntu のインストール準備

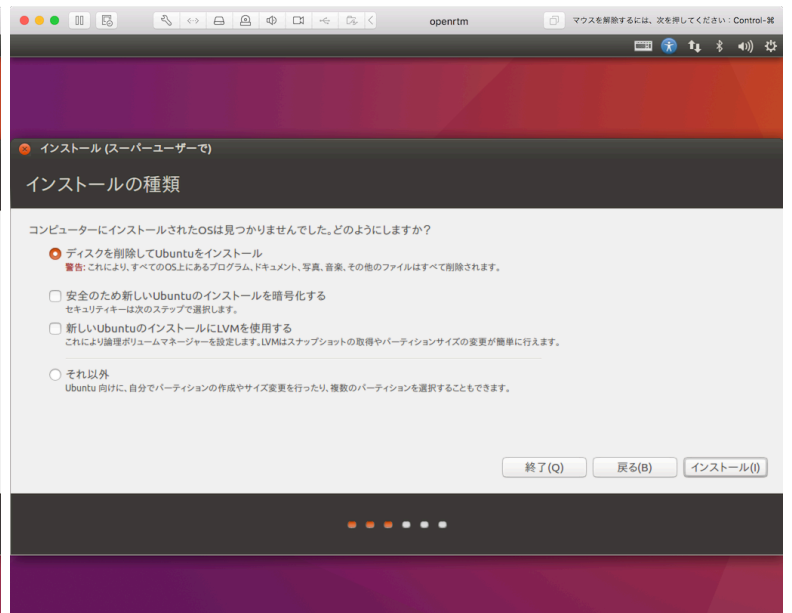


図 8 インストールの種類

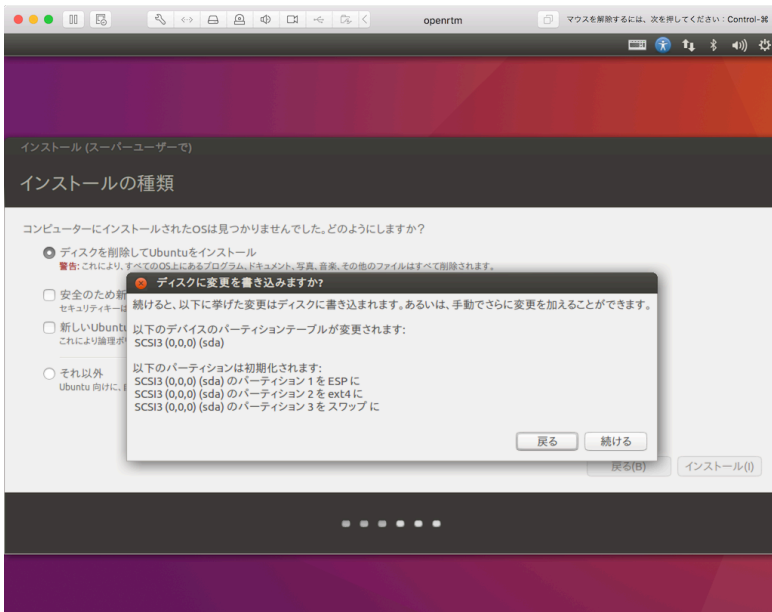


図9 ディスクに変更を書き込みますか？

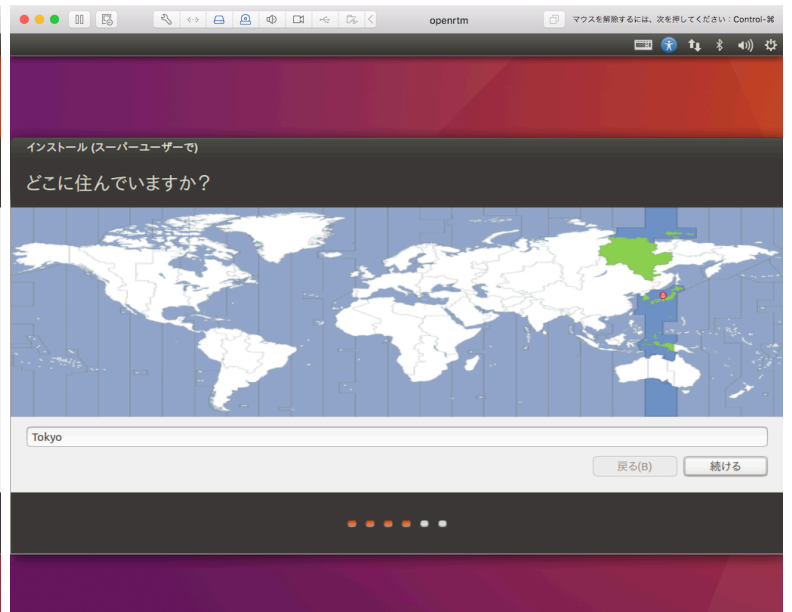


図10 どこに住んでいますか？

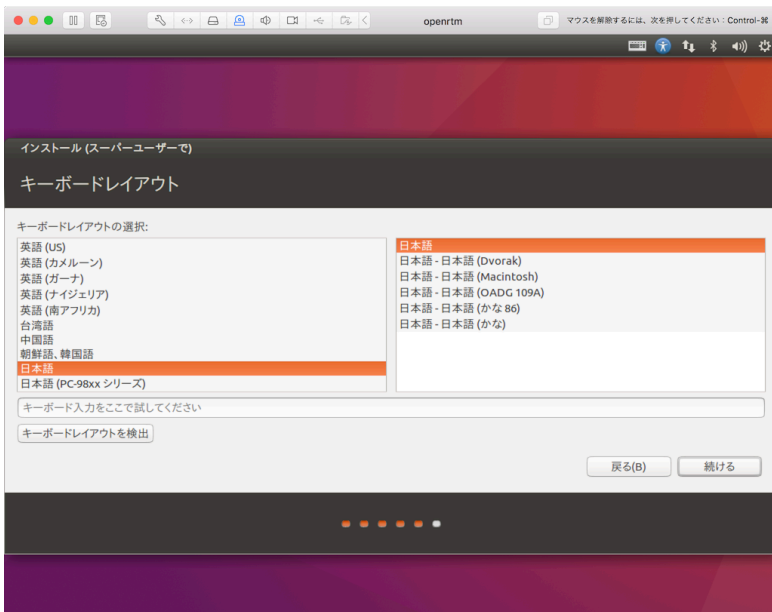


図11 キーボードレイアウト



図12 あなたの情報を入力してください

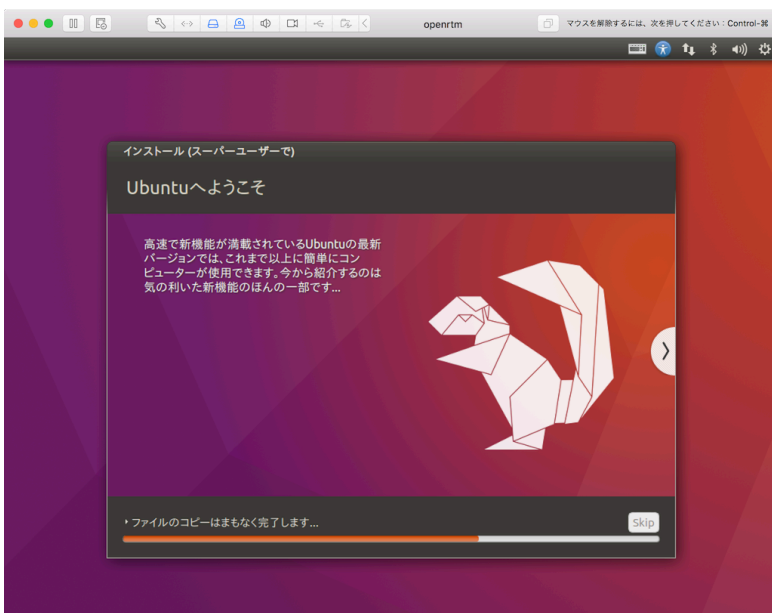


図13 Ubuntuへようこそ

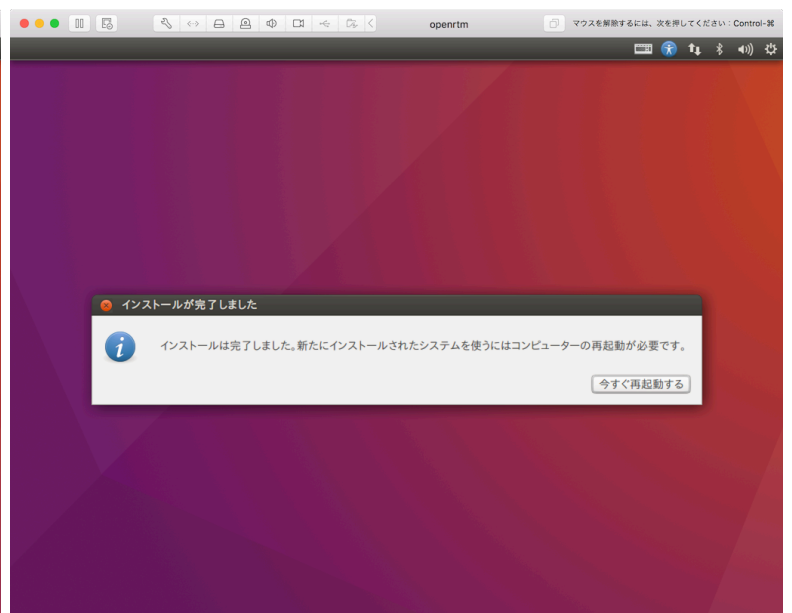


図14 インストール完了しました

## 2 Ubuntu16.04 の設定

### 2.1 インテル® NUC キット NUC7i7BNH の有線 LAN ドライバーをインストール

このままだと有線 LAN・無線 LAN が機能していないため、有線 LAN ドライバーをインストールする。

- <https://downloadcenter.intel.com/ja/product/36773/-> にアクセスし、インテル® PCIe\* インテル® ギガビットイーサネット・ネットワーク・コネクション (Linux\*) 用ネットワーク・アダプター・ドライバーダウンロードページに移動して、`e1000e.x.x.x.tar.gz` をダウンロードして、1.2.1 節で用意した USB メモリに保存する。
- インテル® NUC キットの電源を入れて、Ubuntu にログインし、ドライバを保存した USB メモリを NUC に指す。
- USB メモリにある `e1000e.x.x.x.tar.gz` を、ホームディレクトリにコピーする。
- 右クリックから「端末を開く」をクリックし、以下のコマンドを入力することで、有線 LAN 環境が利用できるようになる。(執筆者環境では `e1000e-3.4.0.2.tar.gz` で行っています。)

```
1 $ tar xvzf e1000e-3.4.0.2.tar.gz
2 $ cd e1000e-3.4.0.2.tar.gz/src
3 $ sudo make install
4 $ sudo modprobe e1000e
5 $ sudo reboot
```

### 2.2 インテル® NUC キット NUC7i7BNH の無線 LAN ドライバーをインストール

- 端末に以下の入力を行い、リポジトリ・ソフトウェアのアップデートを行う。

```
1 $ sudo mv /etc/apt/apt.conf.d/50appstream /etc/apt/apt.conf.d/50
   appstream.disable
2 $ sudo apt update
3 $ sudo apt upgrade
4 $ sudo mv /etc/apt/apt.conf.d/50appstream.disable /etc/apt/apt.conf.d/50
   appstream
5 $ sudo apt-get update
```

[Y/n] で訪ねられたら、「Y」キー押して、「Enter」キーを押す。

- 端末に以下の入力を行うことで、低遅延カーネル (Real-Time Kernel) を適用し、無線 LAN, bluetooth 環境が使用可能になる。

```
1 $ sudo apt install linux-image-lowlatency-hwe-16.04
2 $ sudo reboot
```

### 2.3 「システムプログラムの問題が見つかりました」のダイアログを消す

端末に以下の入力を行うことで、不定期に画面上に出る「システムプログラムの問題が見つかりました」ダイアログを消すことができる。

```
1 $ sudo rm -rf /var/crash/*
```

## 2.4 MATE 環境の導入

端末に以下の入力を行うことで、MATE 環境が使用可能になる。

```
1 $ sudo apt install ubuntu-mate-desktop
2 $ sudo apt remove unity ubuntu-desktop
3 $ sudo apt autoremove
4 $ sudo apt reboot
```

## 2.5 日本語名フォルダを英語名フォルダにする

MATE 端末に以下の入力を行い、ダイアログが出るので、「Update Names」をクリックする。

```
1 $ LANG=C xdg-user-dirs-gtk-update
```

再起動すると、Configured directory for incoming files does not exists ダイアログが出るので、以下のコマンドを実行して、Bluetooth Manager の設定を変更する。

```
1 $ gsettings set org.bluelan.transfer shared-path '/home/e2/Downloads/'
```

# 3 OpenRTM-aist のインストール

## 3.1 必要なソフトウェアのインストール

端末に以下の入力を行う。

```
1 $ sudo apt install python-yaml git
2 $ sudo apt install libomniorb4-dev omniidl omniorb-nameserver
3 $ sudo apt install python-omniorb-omg omniidl-python
4 $ sudo apt install cmake doxygen
5 $ sudo apt install default-jdk
```

## 3.2 OpenRTM のインストール

### 3.2.1 C++ 版のインストール

端末に以下の入力を行う。

```
1 $ wget http://tmp.openrtm.org/pub/Linux/ubuntu/dists/xenial/main/binary-
  amd64/openrtm-aist_1.1.2-0_amd64.deb
2 $ wget http://tmp.openrtm.org/pub/Linux/ubuntu/dists/xenial/main/binary-
  amd64/openrtm-aist-example_1.1.2-0_amd64.deb
3 $ wget http://tmp.openrtm.org/pub/Linux/ubuntu/dists/xenial/main/binary-
  amd64/openrtm-aist-dev_1.1.2-0_amd64.deb
4 $ sudo dpkg -i openrtm-aist_1.1.2-0_amd64.deb
5 $ sudo dpkg -i openrtm-aist-example_1.1.2-0_amd64.deb
```



```
6 $ sudo dpkg -i openrtm-aist-dev_1.1.2-0_amd64.deb
```

### 3.2.2 Python 版のインストール

端末に以下の入力を行う。

```
1 $ wget http://tmp.openrtm.org/pub/Linux/ubuntu/dists/xenial/main/binary-  
  amd64/openrtm-aist-python_1.1.2-1_amd64.deb  
2 $ wget http://tmp.openrtm.org/pub/Linux/ubuntu/dists/xenial/main/binary-  
  amd64/openrtm-aist-python-example_1.1.2-1_amd64.deb  
3 $ sudo dpkg -i openrtm-aist-python_1.1.2-1_amd64.deb  
4 $ sudo dpkg -i openrtm-aist-python-example_1.1.2-1_amd64.deb
```

### 3.2.3 RTSystemEditor/RTCBuilder のインストール

端末に以下の入力を行う。

```
1 $ wget http://tmp.openrtm.org/pub/Linux/ubuntu/dists/xenial/main/binary-  
  amd64/openrtp_1.2.0-0_amd64.deb  
2 $ sudo dpkg -i openrtp_1.2.0-0_amd64.deb
```

端末に以下の入力を行い、必要ない deb ファイルを削除する。

```
1 $ rm -f *.deb
```

## 4 Kinova SDK JACO2 のインストール

Jaco Arm をプログラムで動かすため、Kinova SDK JACO2 をインストールする。

- <https://www.kinovarobotics.com/en/knowledge-hub/all-kinova-products> にアクセスし、サイト内の Software の Kinova SDK JACO<sup>2</sup> をクリックし、KINOVA SDK JACO2\_1.4.2.zip をホームディレクトリにダウンロードする。
- 端末に以下の入力を行い、SDK をインストールする

```
1 $ unzip -d kinova KINOVA SDK JACO2_1.4.2.zip  
2 $ cd kinova/Ubuntu/16_04/64 bits  
3 $ sudo chmod 744 SDK_JACO2_1.4.2_install64  
4 $ sudo sh installSDK64.sh
```

- インストールダイアログが出るので、以下のようにする。
  - 「次へ」をクリックする。(図 15)
  - /opt/JACO2SDK から/opt/JACO2\_SDK に変更して、「次へ」をクリックする。(図 16)
  - 「次へ」をクリックする。(図 17)
  - 「ライセンスに同意する」を選択し、「次へ」をクリックする。(図 18)
  - 「インストール」をクリックする。(図 19)
  - 「完了」をクリックする。(図 20)



図 15 セットアップ - Jaco 2 SDK 画面



図 16 インストール先フォルダ

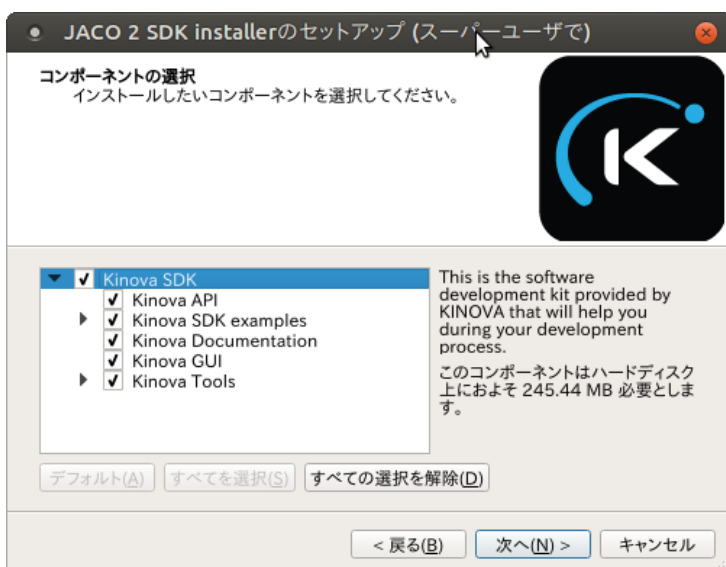


図 17 コンポーネントの選択



図 18 ライセンス条項の同意



図 19 インストールの準備完了



図 20 JACO 2 SDK のウィザードの完了

## 5 PS4 コントローラの設定

低遅延カーネル (Real-Time Kernel) 下では, PS4 コントローラが機能しないため, ds4drv を導入する.

### 5.1 ds4drv のインストール

端末に以下の入力を行う.

```
1 $ sudo apt install python-pip
2 $ sudo pip install ds4drv
```

### 5.2 ds4drv の設定

端末に以下の入力を行う.

```
1 $ git clone https://github.com/chrippa/ds4drv.git
2 $ cd ds4drv/udev
3 $ sudo cp 50-ds4drv.rules /etc/udev/rules.d/
4 $ sudo udevadm control --reload-rules
5 $ sudo udevadm trigger
6 $ cd ../systemd
7 $ pluma ds4drv.service
```

ds4drv.service の 7 行目の「/usr/bin/ds4drv」を, 「/usr/local/bin/ds4drv」に書き換えて保存する.

端末に以下の入力を行う.

```
1 $ sudo cp ds4drv.service /etc/systemd/system/
2 $ sudo systemctl daemon-reload
3 $ sudo systemctl start ds4drv.service
4 $ sudo systemctl enable ds4drv.service
```