

RaspberryPi活用6①

Raspbianの起動

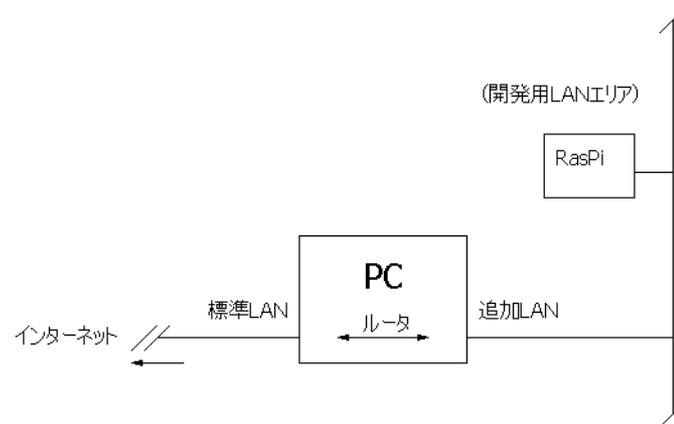
- ・支援ツールとしてWindowsマシン上にLinuxマシンをVMwareWorkstationを介したubuntuを用いる。「付録」のVMwareWorkstationとubuntuのインストール参照。
- ・RaspberryPi本体に使用するOSはRaspbianをベースとする。
- ・RaspberryPiハードウェアはRaspberryPi3 ModelB+

(1. 4GHz/ARM-64bit)



Raspbianの起動

RaspberryPiはPCに接続されたLAN(グローバル側)とは別に以下のような追加LAN環境(ローカル側)で操作する。



Raspbianの起動

<https://www.raspberrypi.org/downloads/> からNOOBSをダウンロードする。



[Blog](#)

[Downloads](#)

[Community](#)

[Help](#)

[Forums](#)

[Education](#)



DOWNLOADS

Raspbian is our official operating system for **all** models of the Raspberry Pi.

Download it here, or use **NOOBS**, our easy installer for Raspbian and more.



NOOBS



RASPBIAN

Raspberry Pi Desktop (for PC and Mac)

Raspbianの起動

2019.6月現在のNOOBSは NOOBS_v3_0_1.zip となっている。

[Blog](#)[Downloads](#)[Community](#)[Help](#)[Forums](#)[Education](#)[Projects](#)

NOOBS

Beginners should start with NOOBS – New Out Of the Box Software. You can purchase a pre-installed NOOBS SD card from many retailers, such as [Pimoroni](#), [Adafruit](#) and [The Pi Hut](#), or download NOOBS below and follow the [software setup guide](#) and [NOOBS setup guide video](#) in our help pages.

NOOBS is an easy operating system installer which contains [Raspbian](#) and [LibreELEC](#). It also provides a selection of alternative operating systems which are then downloaded from the internet and installed.

NOOBS Lite contains the same operating system installer without Raspbian pre-loaded. It provides the same operating system selection menu allowing Raspbian and other images to be downloaded and installed.



NOOBS

Offline and network install

Version: 3.0.1

Release date: 2019-04-08

[Download Torrent](#)[Download ZIP](#)

NOOBS Lite

Network install only

Version: 3.0

Release date: 2018-11-16

[Download Torrent](#)[Download ZIP](#)

Raspbianの起動

NOOBS_v3_0_1.zip を適当なフォルダに展開し準備する。

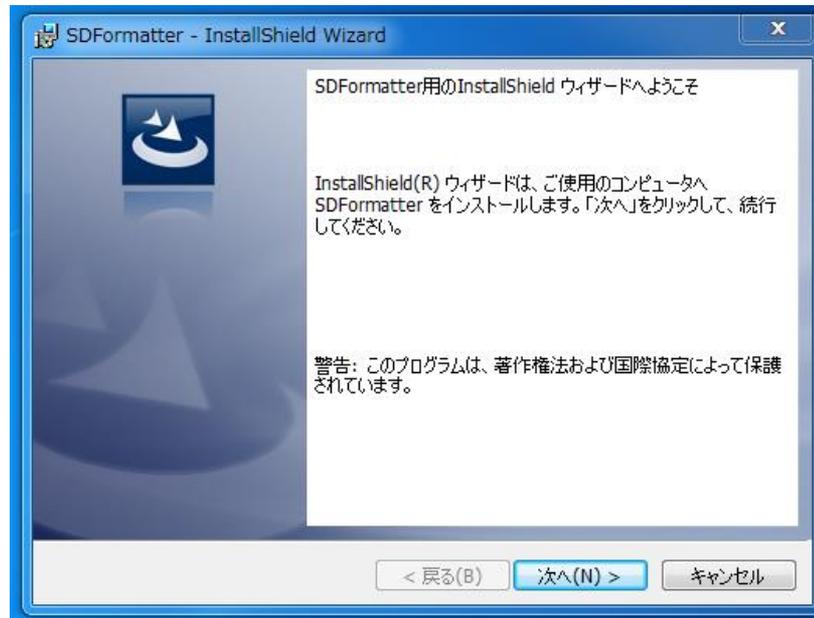
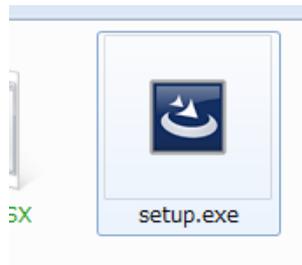


| 名前 | 更新日時 | 種類 | サイズ |
|----------------------------|------------------|---------------|-----------|
| defaults | 2019/05/30 11:49 | ファイル フォル... | |
| os | 2019/05/30 11:49 | ファイル フォル... | |
| overlays | 2019/05/30 11:49 | ファイル フォル... | |
| bcm2708-rpi-0-w.dtb | 2019/05/30 11:49 | DTB ファイル | 23 KB |
| bcm2708-rpi-b.dtb | 2019/05/30 11:49 | DTB ファイル | 23 KB |
| bcm2708-rpi-b-plus.dtb | 2019/05/30 11:49 | DTB ファイル | 23 KB |
| bcm2708-rpi-cm.dtb | 2019/05/30 11:49 | DTB ファイル | 23 KB |
| bcm2709-rpi-2-b.dtb | 2019/05/30 11:49 | DTB ファイル | 24 KB |
| bcm2710-rpi-3-b.dtb | 2019/05/30 11:49 | DTB ファイル | 25 KB |
| bcm2710-rpi-3-b-plus.dtb | 2019/05/30 11:49 | DTB ファイル | 25 KB |
| bcm2710-rpi-cm3.dtb | 2019/05/30 11:49 | DTB ファイル | 24 KB |
| bootcode.bin | 2019/05/30 11:49 | BIN ファイル | 51 KB |
| BUILD-DATA | 2019/05/30 11:49 | ファイル | 1 KB |
| INSTRUCTIONS-README.txt | 2019/05/30 11:49 | TXT ファイル | 3 KB |
| recovery.cmdline | 2019/05/30 11:49 | CMDLINE ファ... | 1 KB |
| recovery.elf | 2019/05/30 11:49 | ELF ファイル | 663 KB |
| recovery.img | 2019/05/30 11:49 | ディスク イメー... | 2,925 KB |
| recovery.rfs | 2019/05/30 11:49 | RFS ファイル | 27,900 KB |
| RECOVERY_FILES_DO_NOT_EDIT | 2019/05/30 11:49 | ファイル | 0 KB |
| recovery7.img | 2019/05/30 11:49 | ディスク イメー... | 2,989 KB |
| riscos-boot.bin | 2019/05/30 11:49 | BIN ファイル | 10 KB |

Raspbianの起動

次に、購入後のSDカードまたは手持ちのSDカードをSDフォーマッタでフォーマットしカードを準備し解凍したファイル類を全部コピーする。

SDフォーマッタについてはSDFormatter4exe.zipをダウンロードし、展開後、setupを実行する。(配布データに収録)



Raspbianの起動

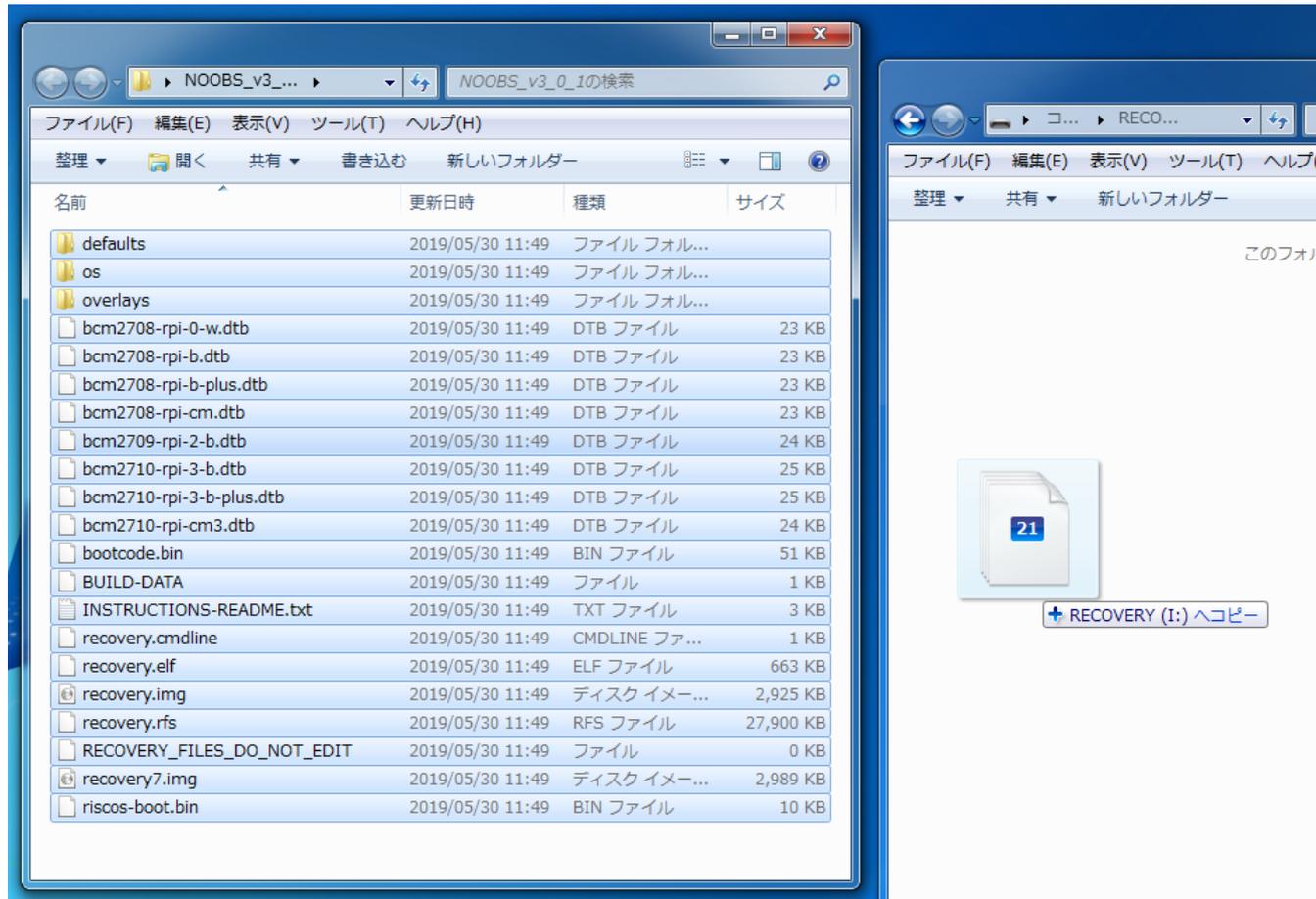
SDFormatterを起動し、フォーマッタでSDカードをフォーマットする。ここではSDカードが I:ドライブに当てられている。「フォーマット」を実行する。



このフォーマットでのSDカードはLinuxは認識しない。ddコマンドを使うためなど認識させたい場合はWindowsアプリでなく、dosアプリのDiskpartによる createparttition primary → active → フォーマット 処理が必要。

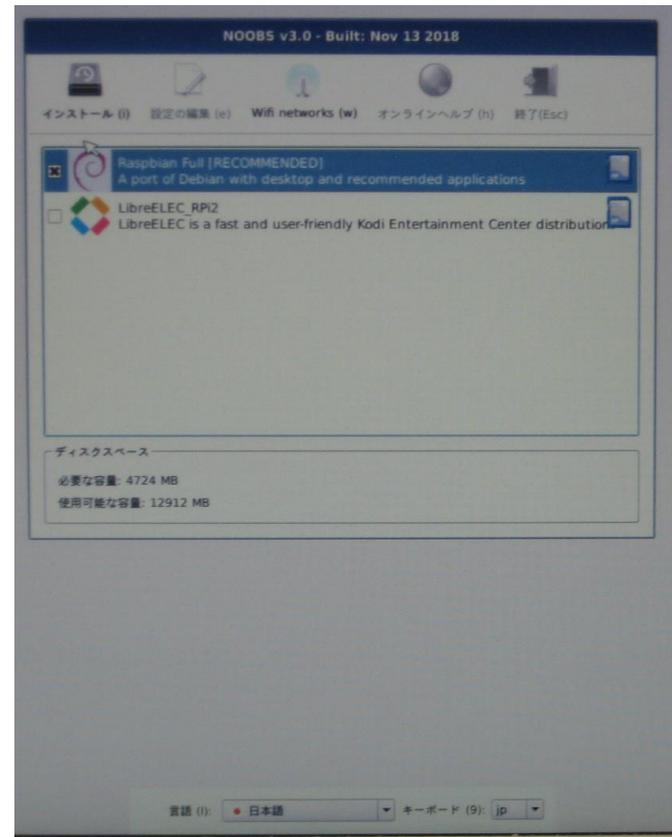
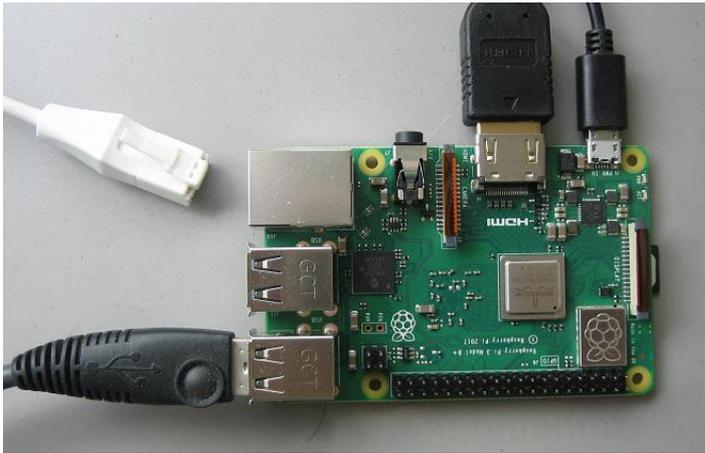
Raspbianの起動

NOOBSの全部をここではI:にコピーする。



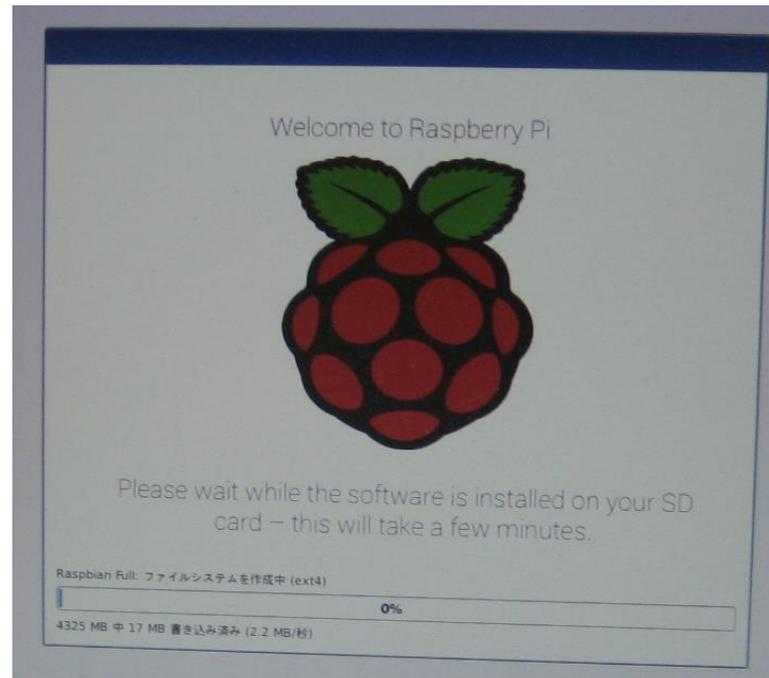
Raspbianの起動

LANケーブルをはずして起動すると、2.4.0以降はRaspbianとLibreELEC_Rpi2のメニューが現れる。Proxy環境なのでLANケーブルをはずして検索時間を省略することができる。(LANケーブルをつないだままだと追加ファイルのインストールを要求されストップする)



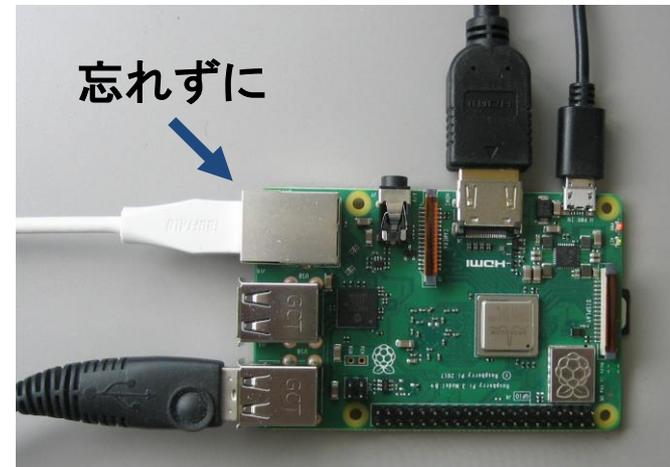
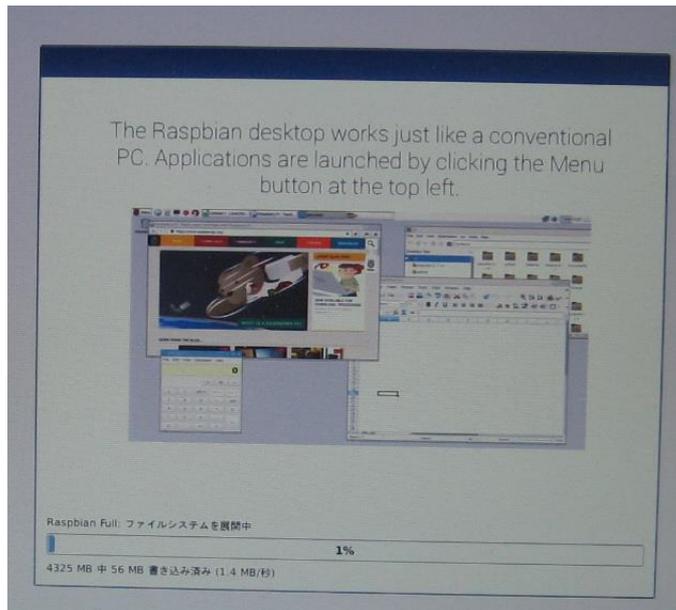
Raspbianの起動

言語を日本語を選択し、Raspbianにチェックを入れ、上部のハードディスクのアイコンをクリックしてインストールする。



Raspbianの起動

約13分(SanDiskUltra48MB/s)～40分(某Class⑩相当)のインストール後「OS(es) Installed Successfully」のメッセージが出る。LANケーブルを接続(これを忘れるとDHCPサーバからのIP割当てができないうえ、その後再割当てが効かないため最初からやり直すことになる)して再起動する。また「付録1」を参照するように、あらかじめVMwarePlayerの環境設定をしてDHCPサーバのあるLAN環境にしておく。



Raspbianの起動

再起動後は、バージョンを重ねるうち初期画面で初期設定ができるようになった。

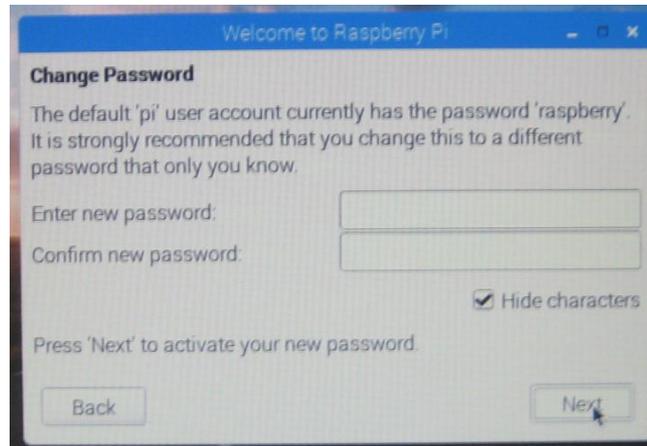


Set Country はJapan,Japanese,Tokyo を選択

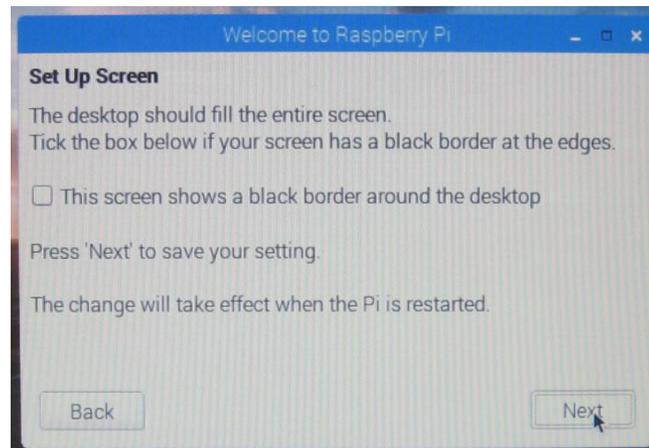


Raspbianの起動

Change Passwordここでは変更せず。記述にあるようにデフォルトのアカウント「pi」、パスワード「raspberrry」となる。

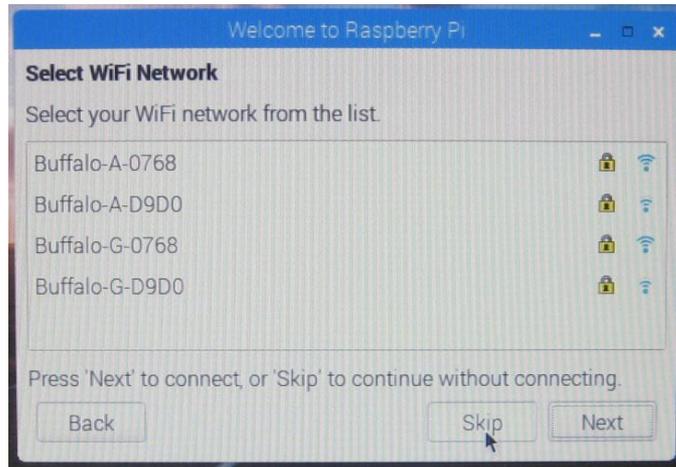


Set Up Screen はそのまま「Next」



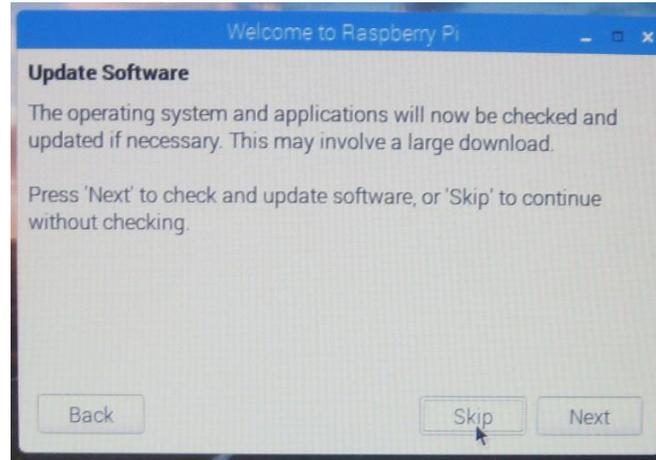
Raspbianの起動

今回はWiFiは使用しないので「Skip」を選択

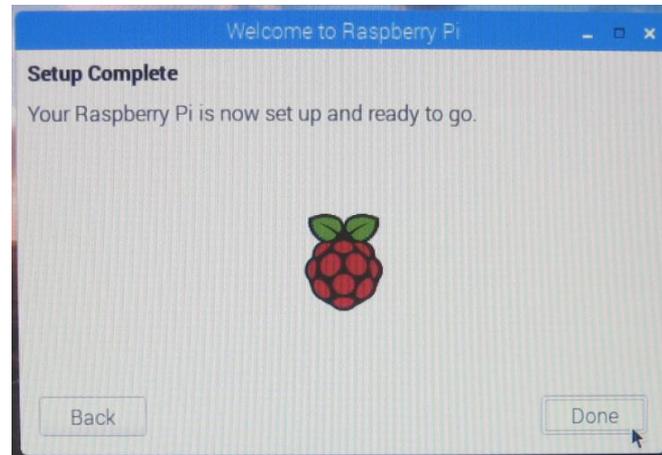


Raspbianの起動

Check For Updates はプロキシ環境ではこの段階で接続できないので「Skip」を選択。

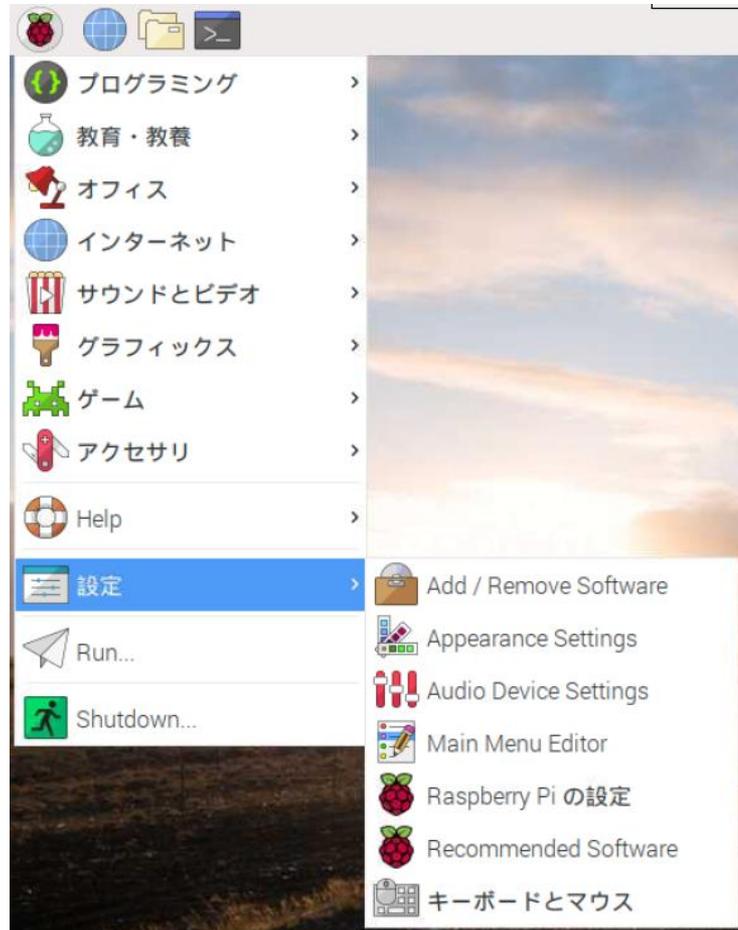


Setup Complete は「Done」を選択。



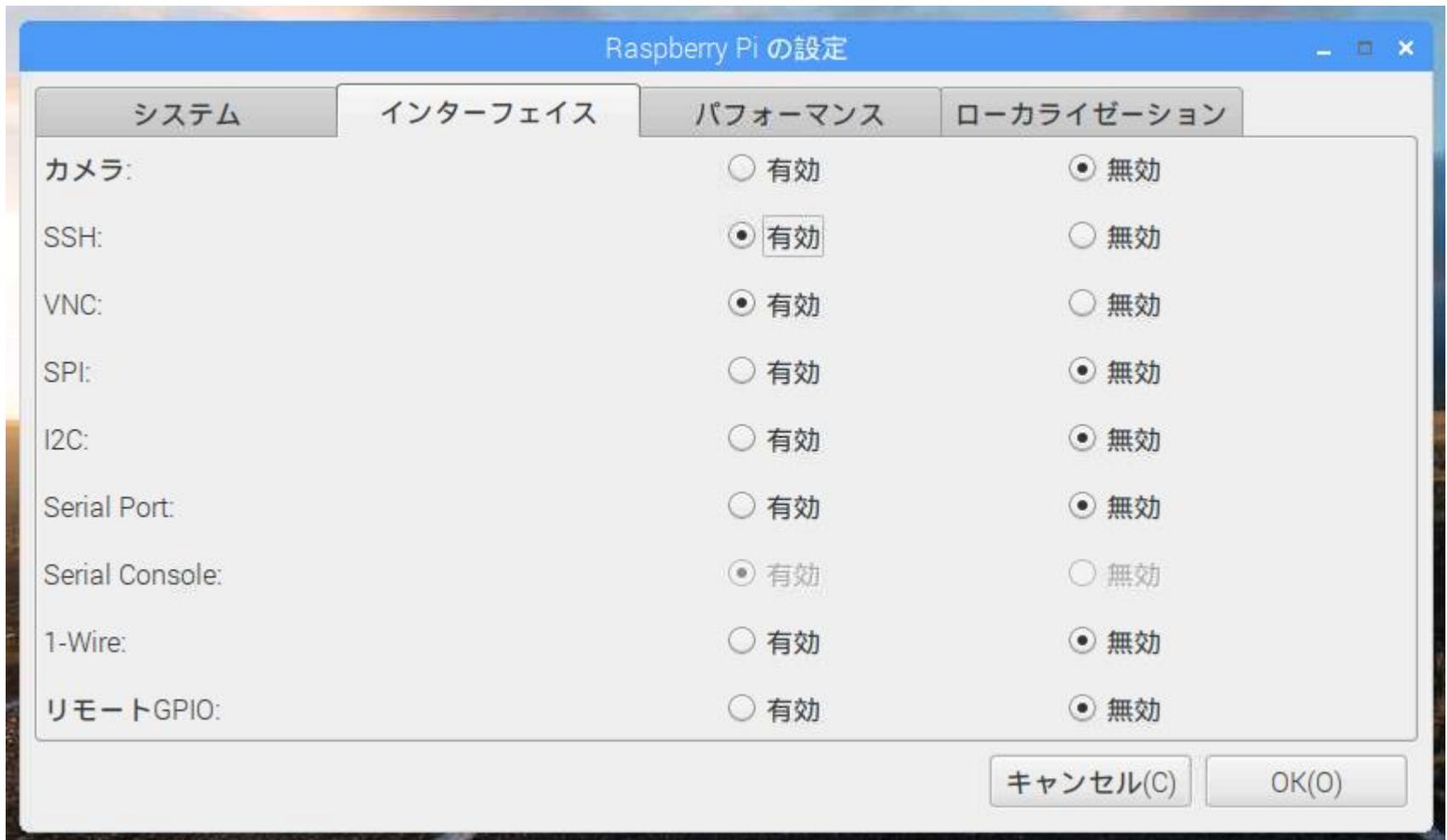
Raspbianの起動

[設定] → [Raspberry Piの設定]を選択



Raspbianの起動

V2.0.0からVNCサーバ機能も追加された。VNCについては後ほど取り上げる。
ここでは[インターフェイス] から「SSH」も有効にする。

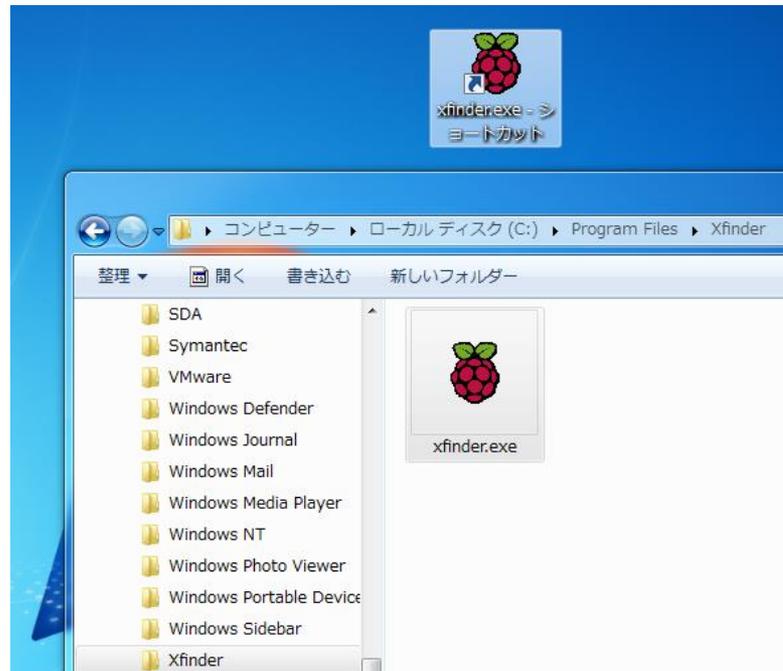


Raspbianの起動

SSHが使える状態になったので、TeraTermとXfinderが使用できる環境を作り、Xfinder.exeを起動する。(VMwareWorkstationをインストールするとDHCPサーバ機能ができる。このためRaspiは同一ネット上に配置される。または別途手動でRaspiをPCマシンと同一ネット上に配置する。)

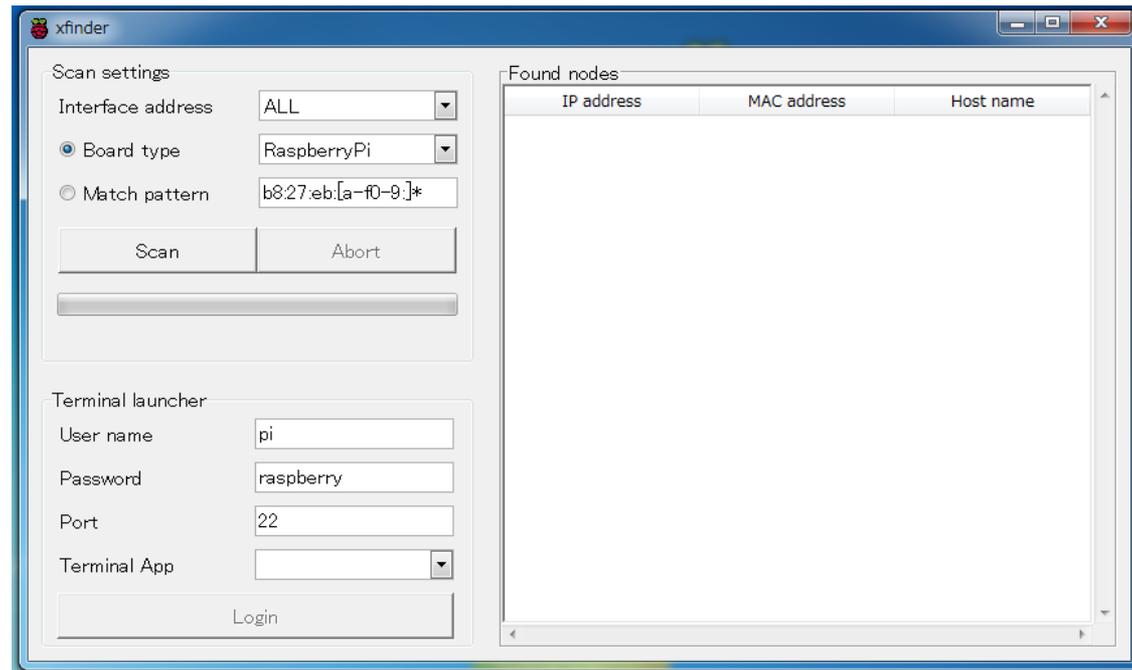
xfinder.exeはネットからダウンロードで入手できる。(配布データに収録)

このアプリにはインストーラはないので適当なフォルダに保存し、このショートカットをデスクトップに作成する。



Raspbianの起動

RaspberryPiをLAN接続し、xfinder.exeを起動する。
デフォルトのターミナルアプリ(TerminalApp)は空白である。



Raspbianの起動

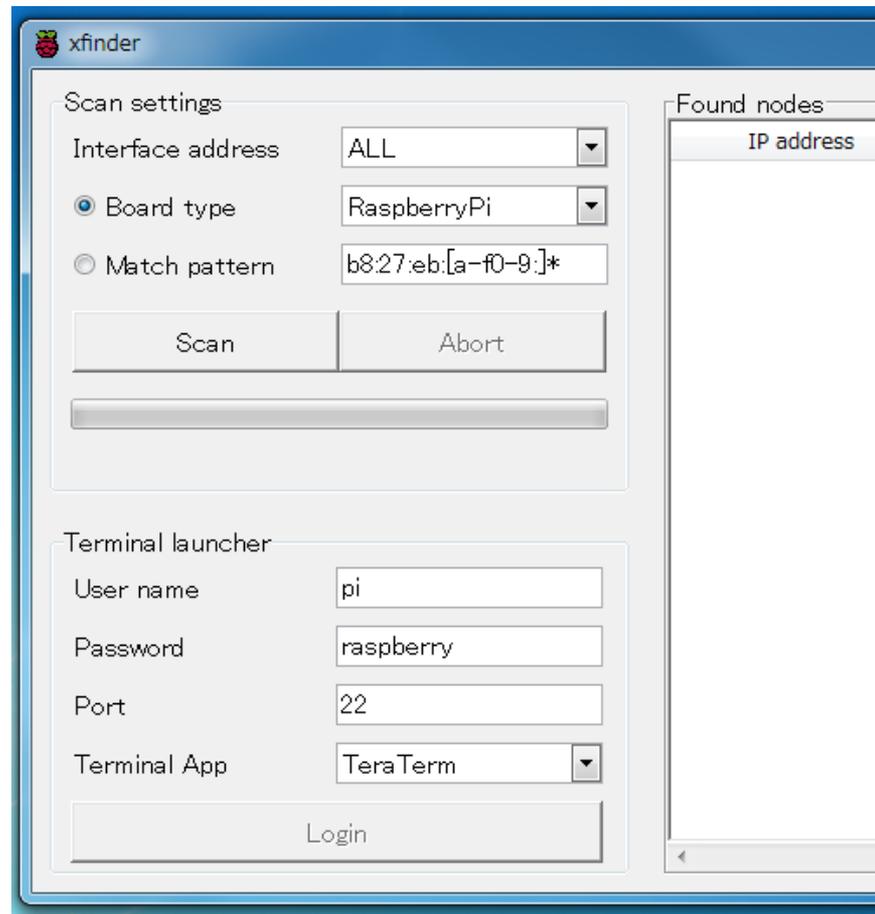
ここではターミナルアプリとしてTeraTermを使用する。

ネットからteraterm-4.100.exeを入手(配布データに収録)、インストールする。



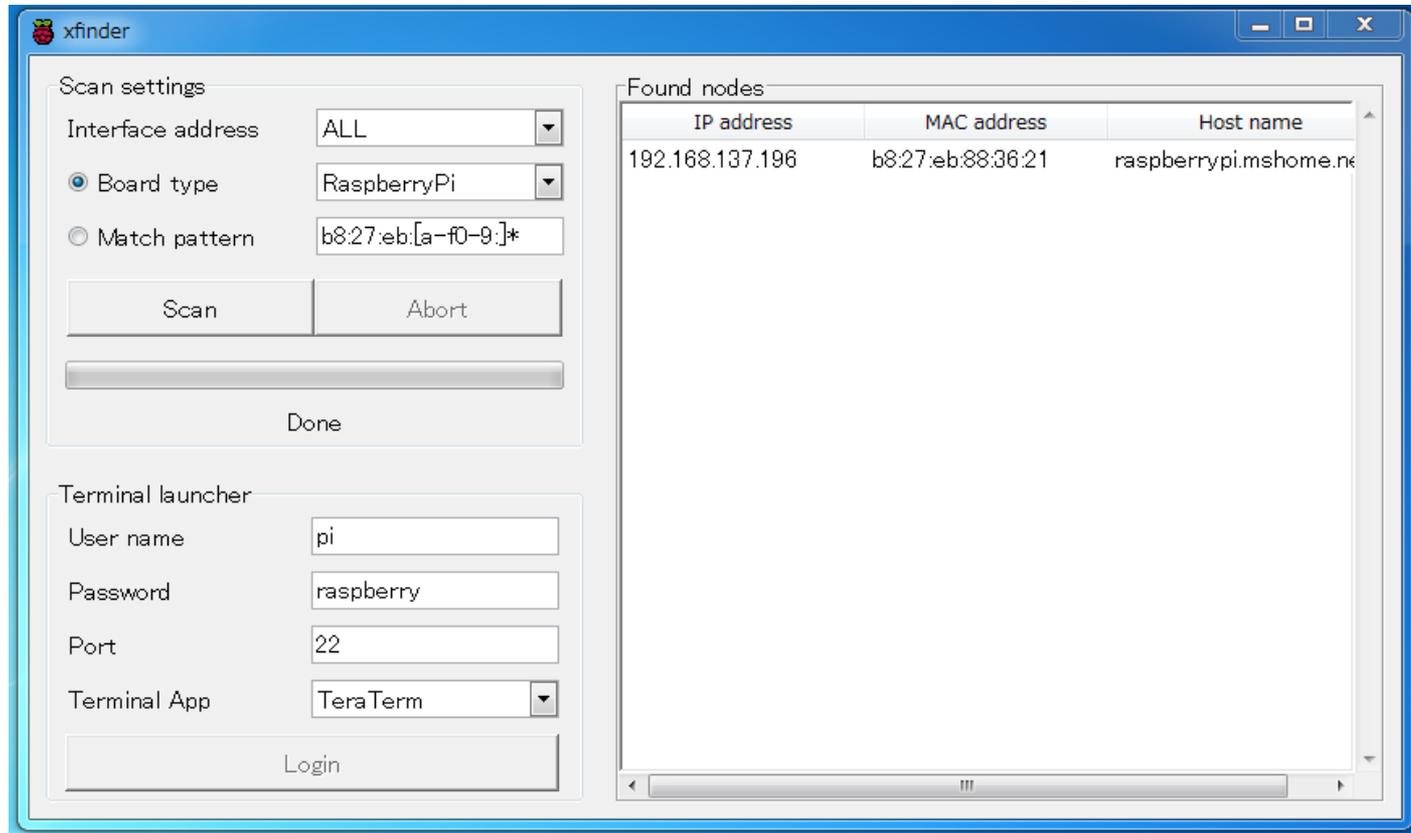
Raspbianの起動

xfinder.exeを起動。あらためてデフォルトのターミナル(TerminalApp)にTeraTermを選択する。Raspberryのデフォルトのユーザネームは「pi」、パスワードは「raspberry」である。



Raspbianの起動

「Scan」をクリックすると、ローカルネットワークにRaspberrypiを見つけることができる。以下例ではIPアドレスが192.168.137.196(この段階ではDHCPサーバが決めた)のものが見つまっている。複数のRasPiがある場合はこの段階では対象外をネットワークからはずして見極める。



Raspbianの起動

以降のため IPアドレスを192.168.137.55 に設定しておく。

RaspberryPiの右上のメニューから右クリック。

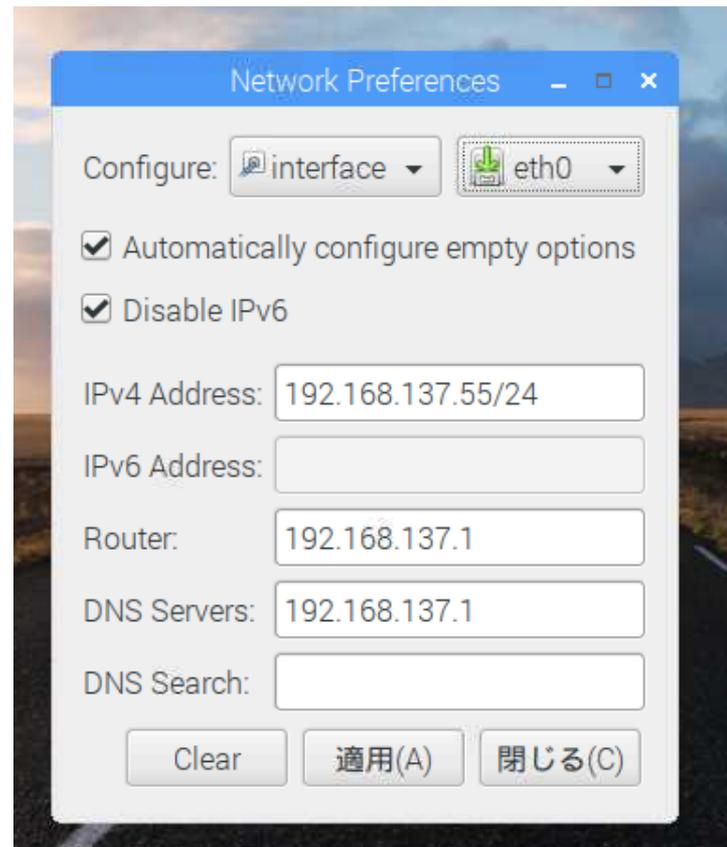
「Wireless & Wired Network Settings」を選択する。



Raspbianの起動

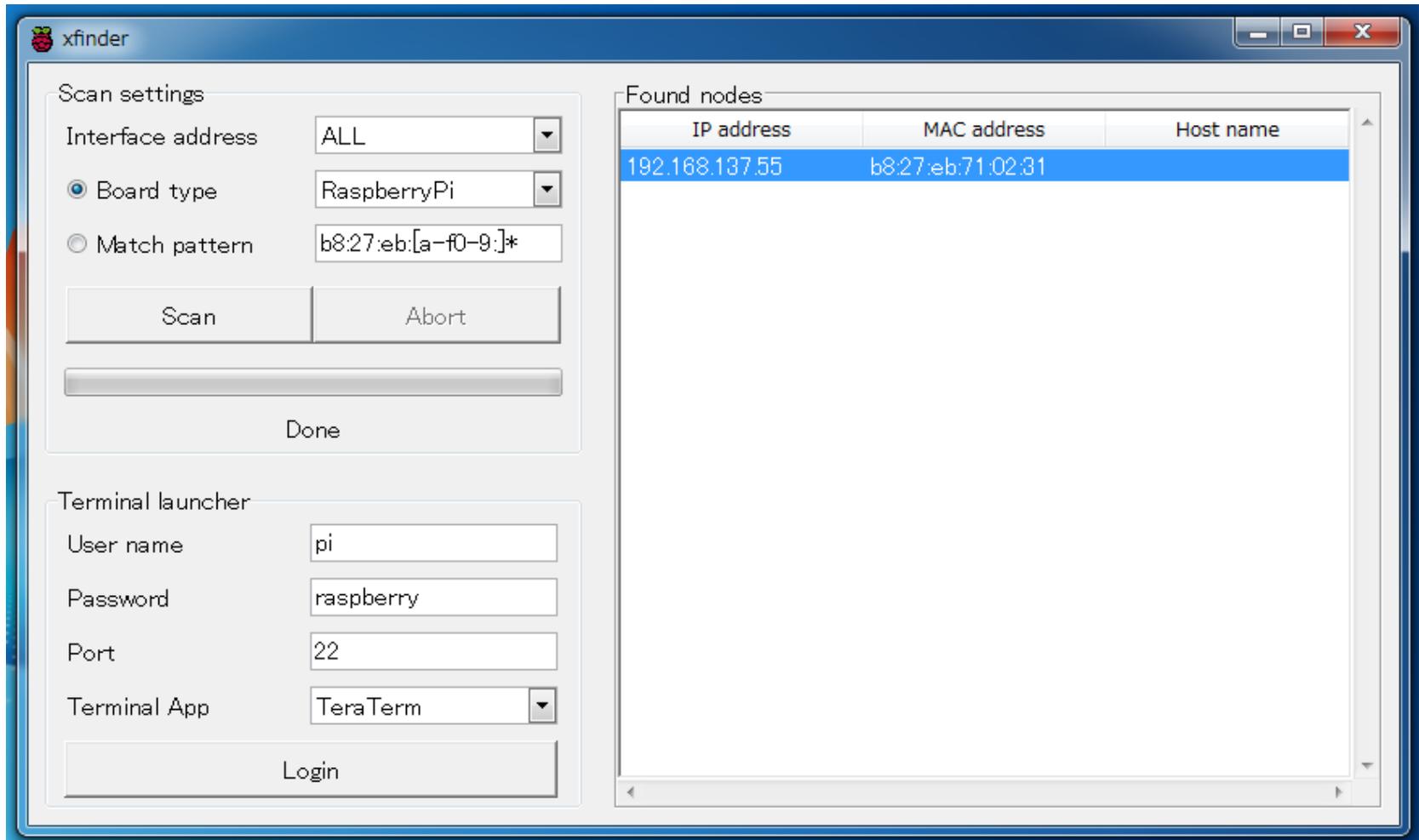
細かな設定説明は以降を参照。

ここでは以下のように Ipv4 Addressを 192.168.137.55 とする。



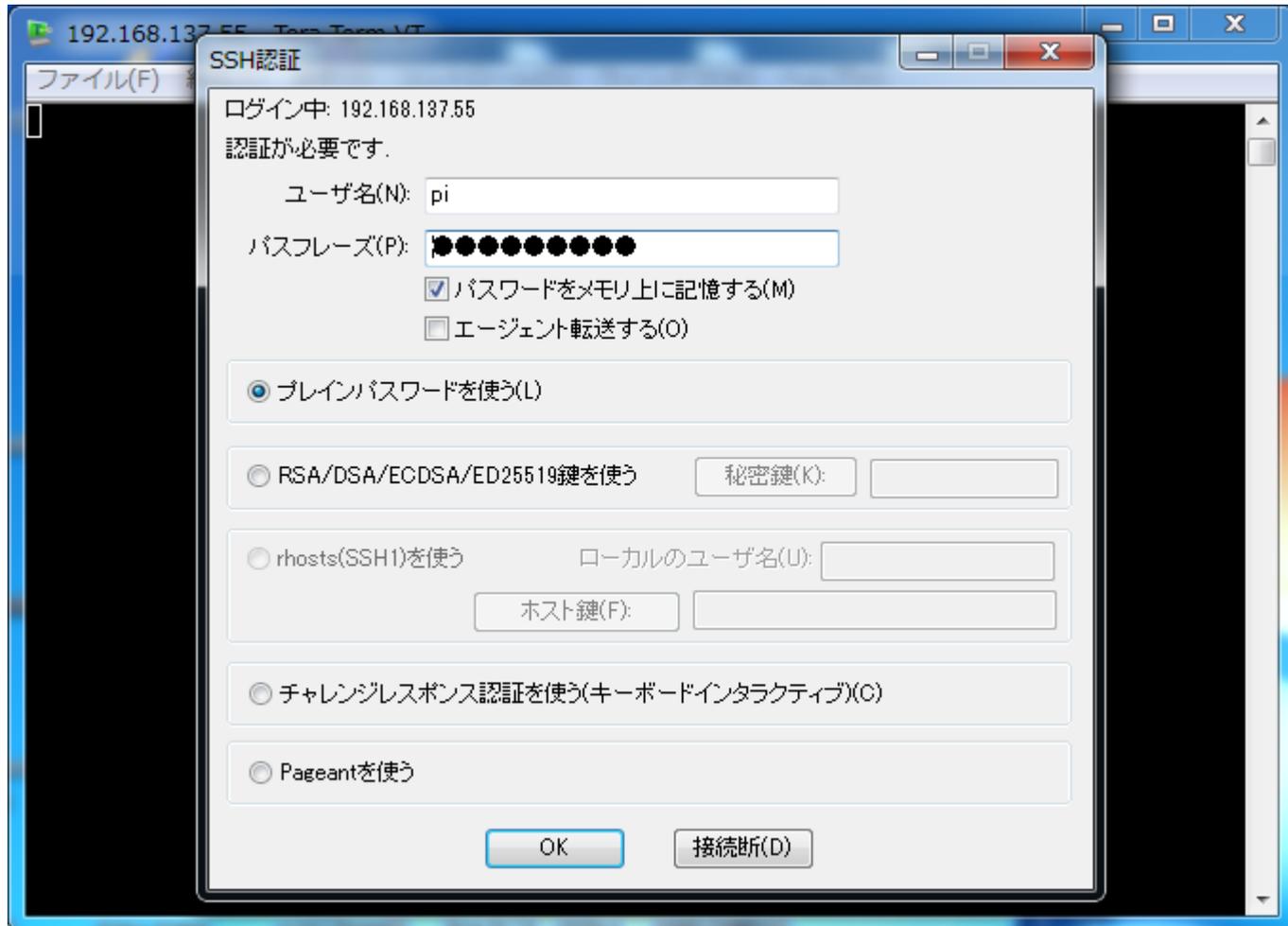
Raspbianの起動

対象のIPアドレスを選んで「Login」をクリック。



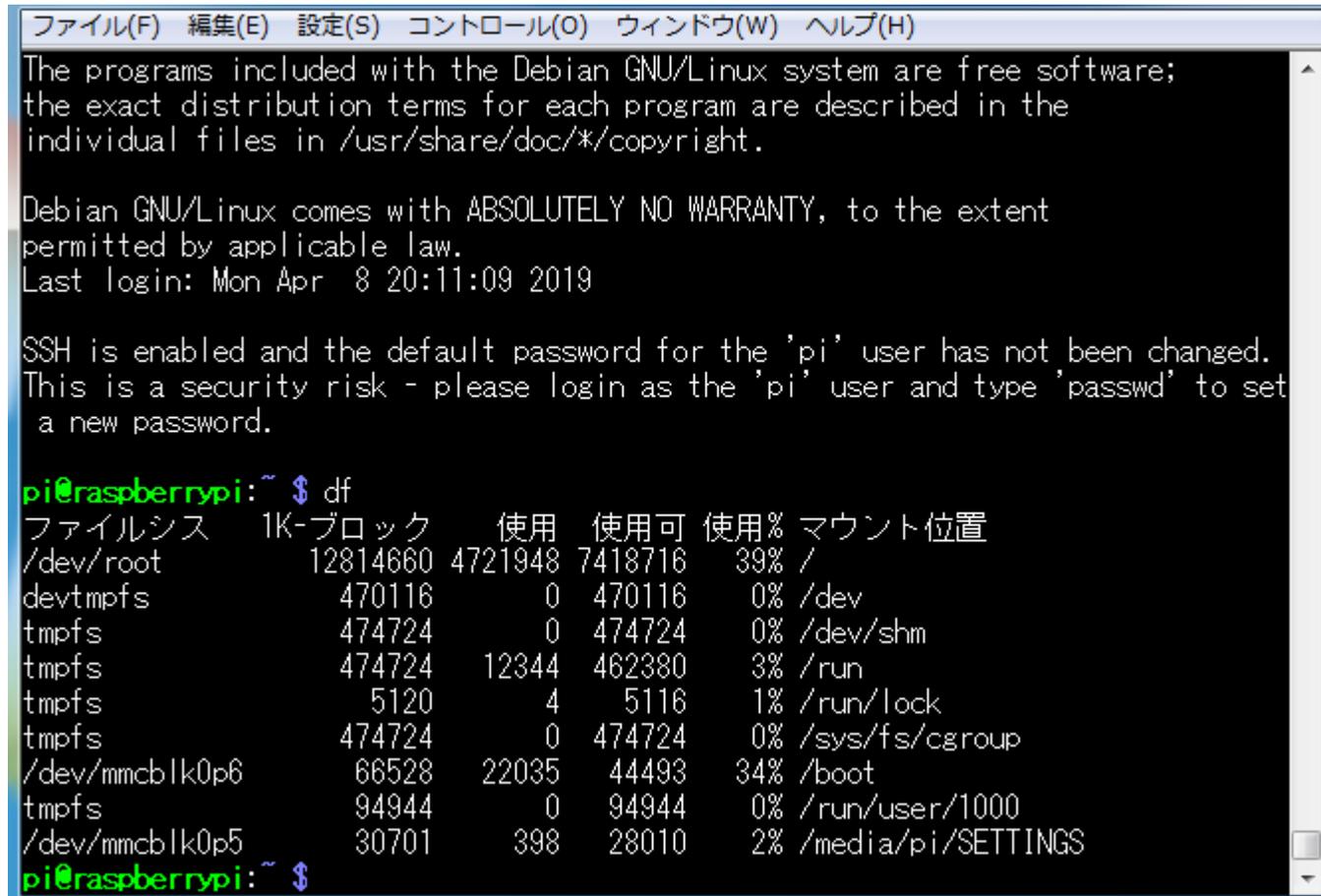
Raspbianの起動

SSHを有効にしているため、セキュリティ警告の「続行」を選択後、以下のようなTeraTerm起動が開く。「OK」を選択すると見慣れたターミナルが現れる。



Raspbianの起動

\$dfにてファイルシステムの確認。ubuntuPCで見たファイルブロック (/SHETTING, /boot ちなみに /rootは現在の自身)がある。



```
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Apr  8 20:11:09 2019

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.

pi@raspberrypi:~ $ df
ファイルシス 1K-ブロック 使用 使用可 使用% マウント位置
/dev/root    12814660 4721948 7418716 39% /
devtmpfs    470116 0 470116 0% /dev
tmpfs       474724 0 474724 0% /dev/shm
tmpfs       474724 12344 462380 3% /run
tmpfs       5120 4 5116 1% /run/lock
tmpfs       474724 0 474724 0% /sys/fs/cgroup
/dev/mmcblk0p6 66528 22035 44493 34% /boot
tmpfs       94944 0 94944 0% /run/user/1000
/dev/mmcblk0p5 30701 398 28010 2% /media/pi/SETTINGS
pi@raspberrypi:~ $
```

Raspbianの起動

プロキシ環境ではインターネット上のタイムサーバーとの時刻同期ができない。同期がとれている場合は時間設定に関する詳細の「付録2」は飛ばしてよい。ここでは簡単に述べておく。

Raspiのntpクライアントプログラムはsystemd-timesyncdが働いており、いつの頃のバージョンからntpのサポートもなくなった。

\$ timedatectl status でプロキシ環境でのデフォルト状態。

```
pi@raspberrypi:~$ timedatectl status
Local time: 月 2019-04-08 20:41:17 JST
Universal time: 月 2019-04-08 11:41:17 UTC
RTC time: n/a
Time zone: Asia/Tokyo (JST, +0900)
Network time on: yes
NTP synchronized: no
RTC in local TZ: no
pi@raspberrypi:~$
```

Raspbianの起動

```
pi@raspberrypi:~ $ timedatectl status
    Local time: 月 2019-04-08 20:41:17 JST
    Universal time: 月 2019-04-08 11:41:17 UTC
    RTC time: n/a
    Time zone: Asia/Tokyo (JST, +0900)
    Network time on: yes
    NTP synchronized: no
    RTC in local TZ: no
pi@raspberrypi:~ $
```

Network time on: yes (no) の切り替えは

`sudo timedatectl set-ntp true` (あるいはfalse) でおこなう。

また、`systemd-timesyncd`によるNTPサーバーの指定方法は
`/etc/systemd/timesyncd.conf` 編集と

`sudo systemctl start (stop, restart) systemd-timesyncd` でおこなう。

NTP synchronized: no (yes) は`systemd-timesyncd`による同期状態でyesに替わる。

プロキシ環境の時間設定の詳細「付録2」を参照。

Raspbianの起動

時刻設定済後、ファイルのアップデートをする。

プロキシ環境で作業する場合は以下の書式でコマンド処理する。

```
$sudo http_proxy=http://(プロキシサーバIP):(ポート) apt-get update
```

```
pi@raspberrypi:~ $ sudo http_proxy=http://172.16.0.8:8080 apt-get update
```

このように apt-get コマンドで毎回、プロキシサーバ指定するのは大変なので /etc/apt/apt.conf ファイルを新規作成し、保存する。

ここではubuntu系エディタ”nano”で編集する。(行末のセミコロンを忘れずに)

```
pi@raspberrypi:~ $ sudo nano /etc/apt/apt.conf
```

```
Acquire::ftp::proxy "ftp://172.16.0.8:8080/";  
Acquire::http::proxy "http://172.16.0.8:8080/";  
Acquire::https::proxy "https://172.16.0.8:8080/";  
Acquire::socks::proxy "socks://172.16.0.8:8080/";
```

Ctrl+O→「enter」、Ctrl+Xで保存と終了。

Raspbianの起動

updateには本施設環境で約2分ほど要する。

次に、ファイルのアップグレードをする。

/etc/apt/apt.conf を作成したので

\$sudo apt-get upgrade として プロキシサーバとポート番号は省略できる。

```
pi@raspberrypi:~$ sudo apt-get upgrade
```

プロキシ環境では完全に終了しないことがある。

```
- MD5Sum:bca538436aa39e34c412287da2086ad6 [weak]
- Filesize:764754 [weak]
E: いくつかのアーカイブを取得できません。apt-get update を実行するか --fix-missing オプションを試してみてください。
pi@raspberrypi:~$
```

Raspbianの起動

- `--fix-missing` オプションでもプロキシサーバ経由だと、いくつかの接続失敗が出てして完全には終了しない。

これらについてはこのままにしておく。

エラー検出の影響で多少遅くなるが、SDカードの性能でも大きく差異が出る。

本施設の環境では、SanDisk Ultra 48MB/s にて約30分以上の処理となるので実施は空いた時間におこなう。

```
pi@raspberrypi:~ $ sudo apt-get upgrade --fix-missing
```

```
E: http://ftp.tsukuba.wide.ad.jp/Linux/raspbian/raspbian/poc-3-1+deb9u6+rpi1\_armhf.deb の取得に失敗しました Writing mode (984810)
```

```
Hashes of expected file:
```

- SHA256:4f7a7db744d52592c3ae7557dc21817c13dbce0fc40b6fc
- SHA1:5e7bd2c8145101c5fca9079e8f67572f3f842bea [weak]
- MD5Sum:6f8d95b01796dab490565bd0f971c26b [weak]
- Filesize:2866580 [weak]

```
pi@raspberrypi:~ $
```

Raspbianの起動

当施設はプロキシ環境なので update,upgrade が完了しないので別途ブロックされない環境で実施したデータを用意してあるので気になる方は以下の処理をおこなう。

/home/pi に _Raspbian_save にデータを入れたとして

```
pi@raspberrypi:~ $ cd _Raspbian_save/  
pi@raspberrypi:~/_Raspbian_save $ ls  
var_cache_apt_archives_3-0-1_190619  var_lib_apt_lists_3-0-1_190619  
pi@raspberrypi:~/_Raspbian_save $ sudo cp var_cache_apt_archives_3-0-1_190619/*  
/var/cache/apt/archives/  
pi@raspberrypi:~/_Raspbian_save $ sudo cp var_lib_apt_lists_3-0-1_190619/*  
/var/lib/apt/lists/  
pi@raspberrypi:~/_Raspbian_save $
```

Raspbianの起動

改めて update する。

```
pi@raspberrypi:~$ sudo apt-get update
ヒット:1 http://archive.raspberrypi.org/debian stretch InRelease
ヒット:2 http://raspbian.raspberrypi.org/raspbian stretch InRelease
パッケージリストを読み込んでいます... 完了
pi@raspberrypi:~$
```

upgrade する。

```
pi@raspberrypi:~$ sudo apt-get upgrade
libreoffice-report-builder-bin (1:5.2.7-1+rpi1+deb9u7) を設定しています ...
libreoffice-report-builder (1:5.2.7-1+rpi1+deb9u7) を設定しています ...
libreoffice (1:5.2.7-1+rpi1+deb9u7) を設定しています ...
initramfs-tools (0.130) のトリガを処理しています ...
libc-bin (2.24-11+deb9u4) のトリガを処理しています ...
libvlc-bin:armhf (3.0.7-0+rpt1+deb9u1) のトリガを処理しています ...
libreoffice-common (1:5.2.7-1+rpi1+deb9u7) のトリガを処理しています ...
pi@raspberrypi:~$
```

完了。メッセージはなくなっている。

Raspbianの起動

VNC機能が標準装備となった。一方、これまでのtightvncserverの機能も捨て難いのでこの2つ接続方法を示す。

tightvncserverは標準VNCと違い、個別にターミナルを複数生成することができたり、デフォルトのウィンドウサイズが指定できる。一方、標準VNCは接続するモニタ出力の解像度のターミナルのコピーの画面が得られる。内蔵カメラ映像やSkype画面はモニタ出力からしか利用できないことが判明しているので内蔵カメラ映像やSkype利用時は標準VNCを使うことになる。

~~また、これら2つのVNCは併用することはできない。~~

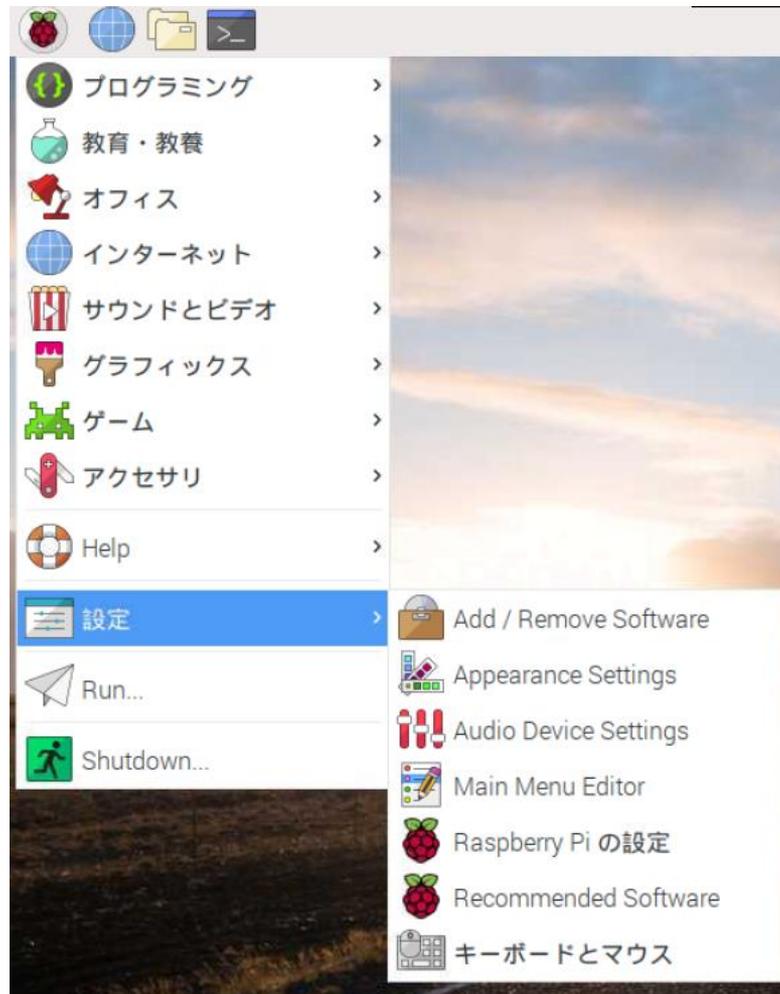
バージョンを重ねるうちこれまでのような tightvncserver は標準VNCを無効にしないと回復不能な暴走もなくなった。

また、単にtightvncserverをアンインストールするだけでは標準VNCは回復できないのは同じであるが、うまく設定さえすると併用も可能になった。(今のところ弊害発生せず)

標準VNCの回復、併用のいずれにせよ tightvncserver を使用する場合にはこれに先立ち、回復に必要なファイルの退避が必要となる。

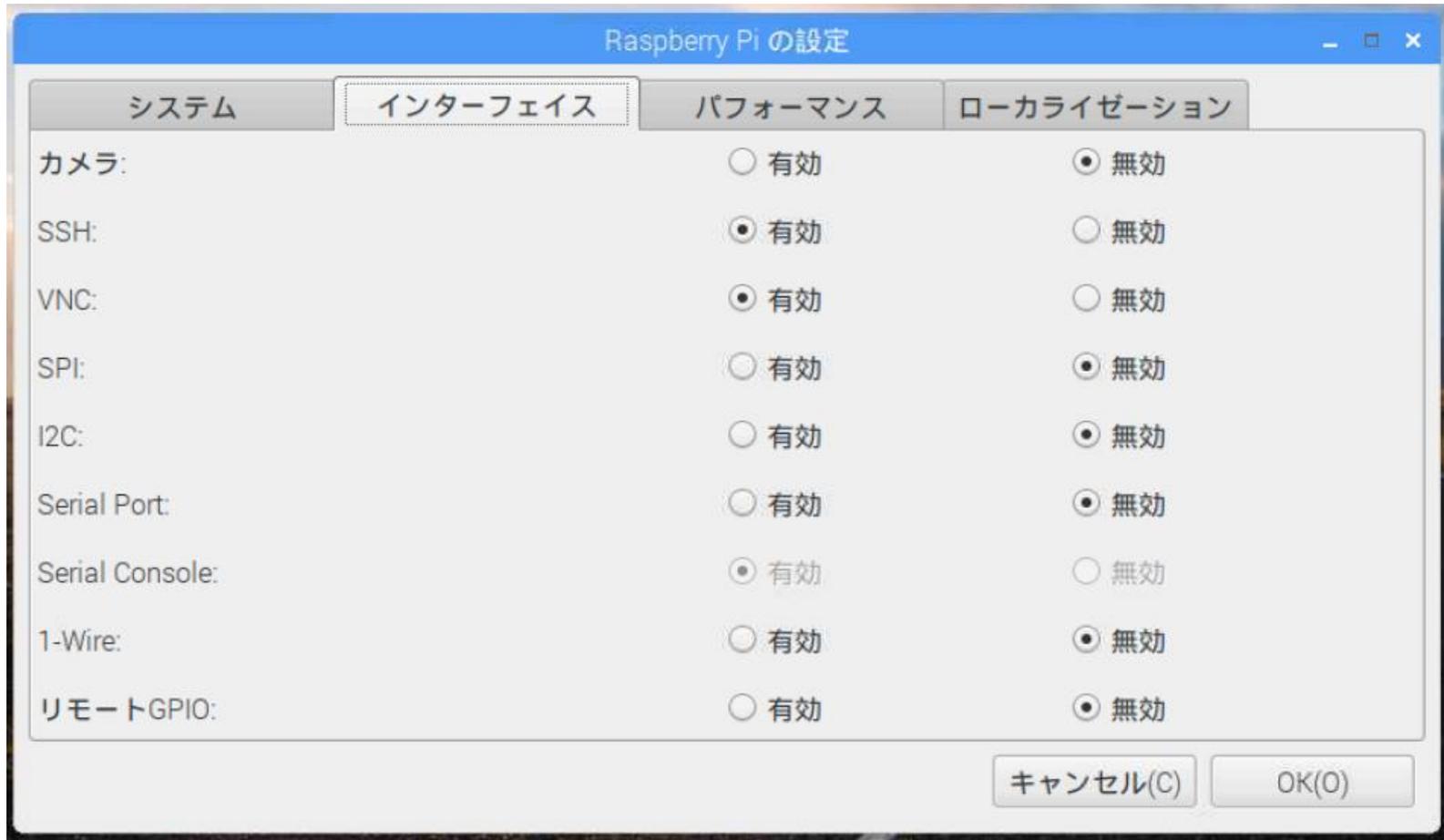
Raspbianの起動

まずは標準VNCの起動について。「設定」→「RaspberryPiの設定」



Raspbianの起動

「インターフェイス」を選択し「VNC」を有効を確認する。

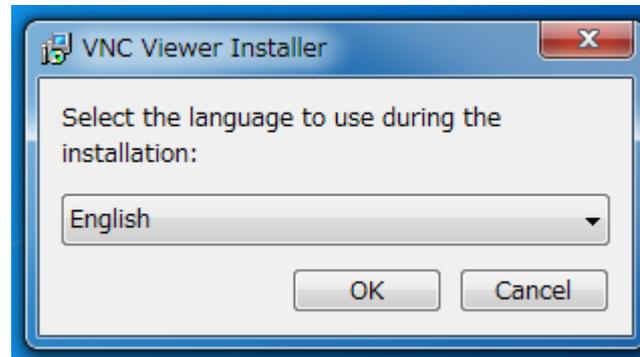
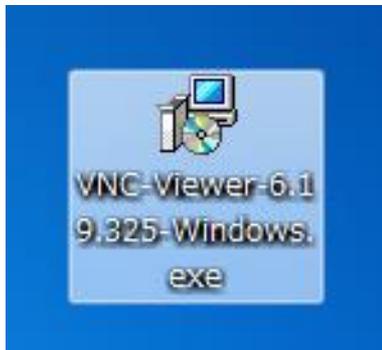


Raspbianの起動

VNCサーバに対し、クライアント側はWindowsで動作するVNC Viewerを使用する。2019.6現在VNC-Viewer-6.19.325。

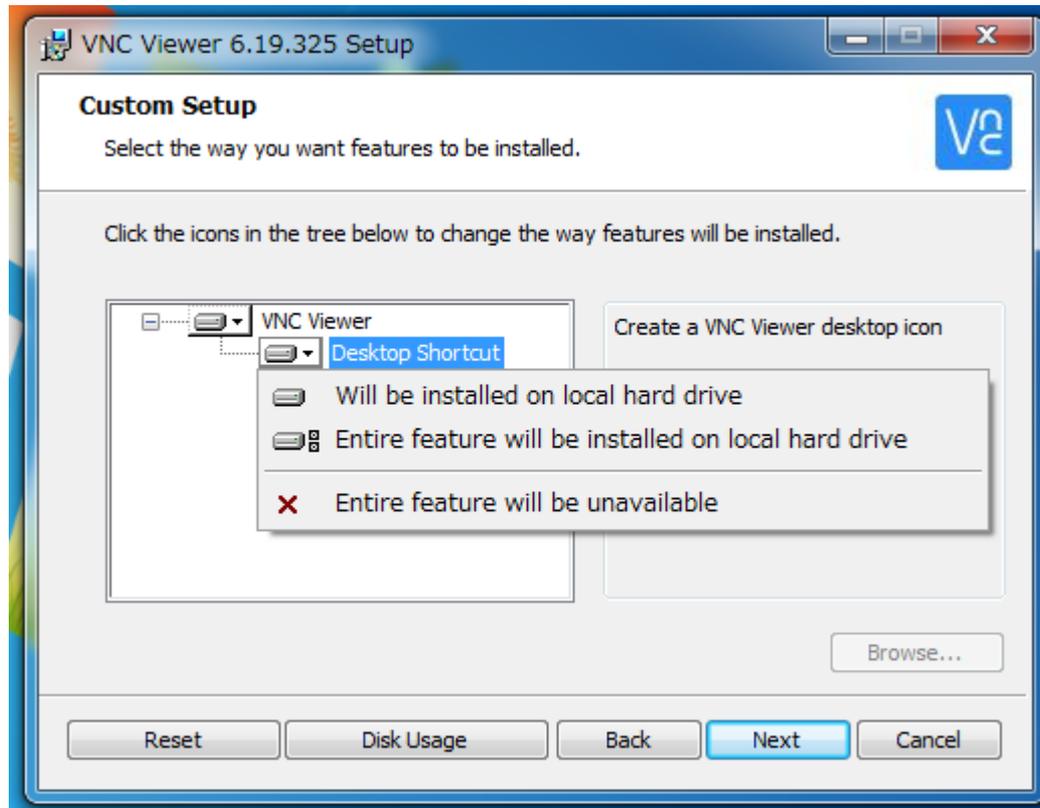
VNC Viewerのインストールはネットからダウンロードできる。
(配布データに収録)

VNC-Viewer-6.18.625-Windows.exeを実行する。



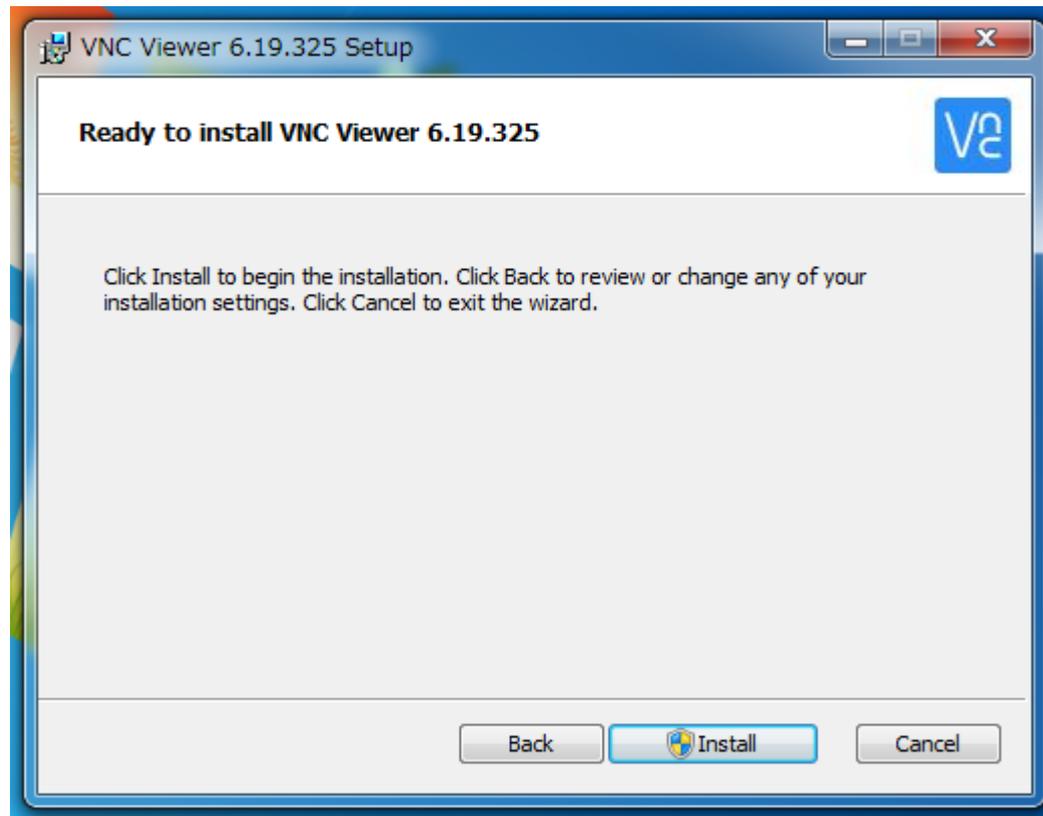
Raspbianの起動

Desktop Shortcutを作成する。



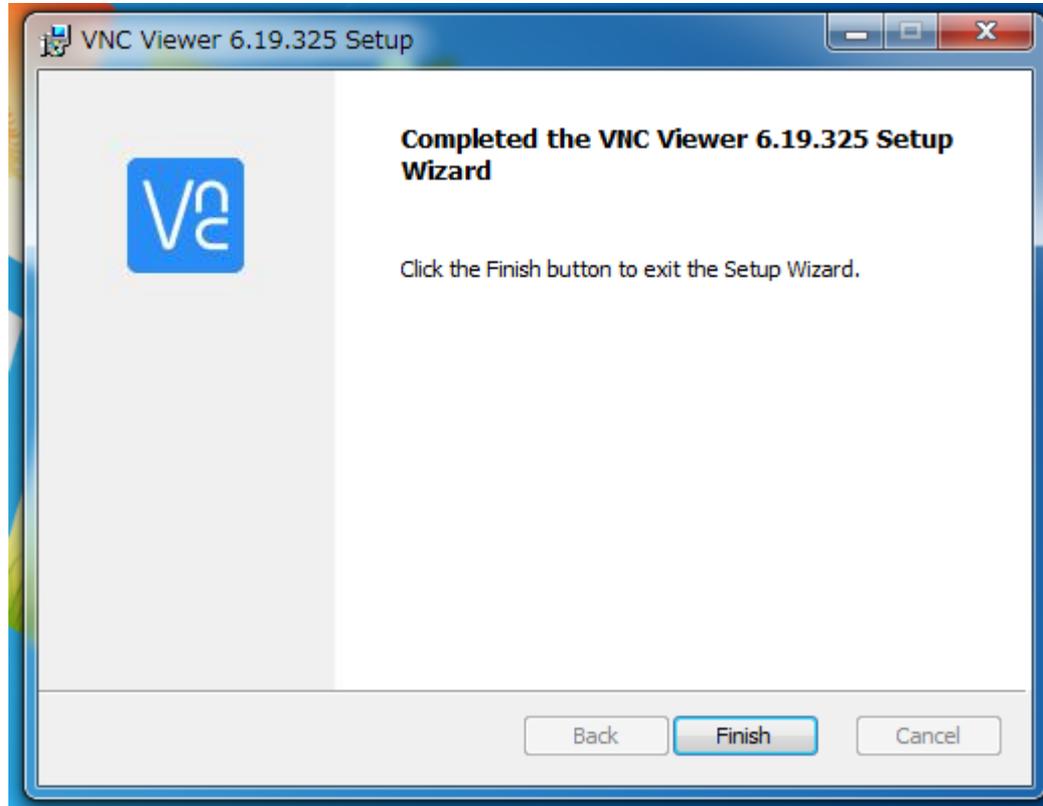
Raspbianの起動

諸設定はデフォルトのまま「Install」。



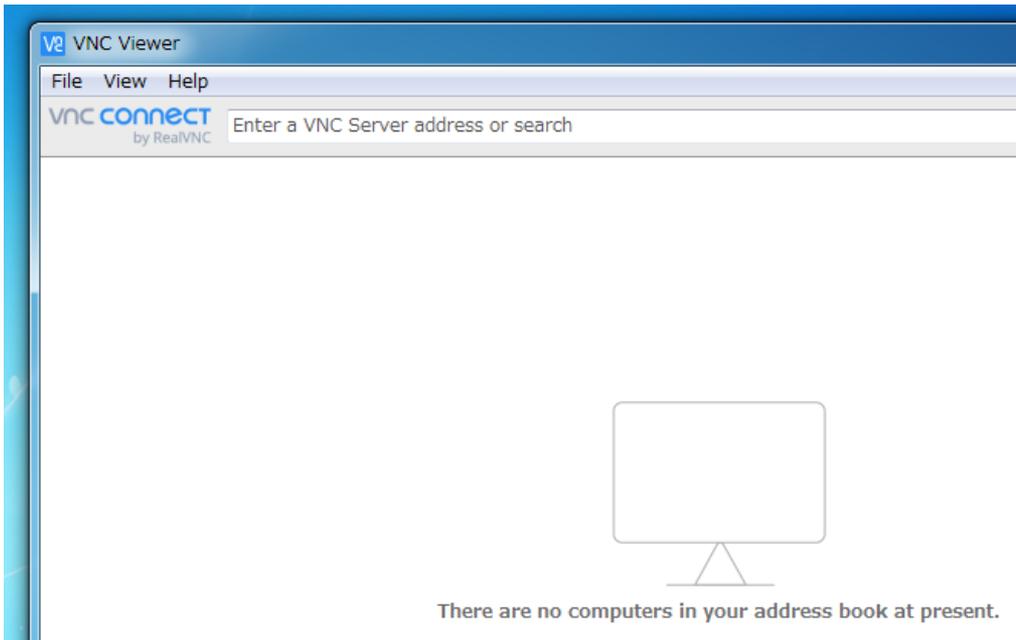
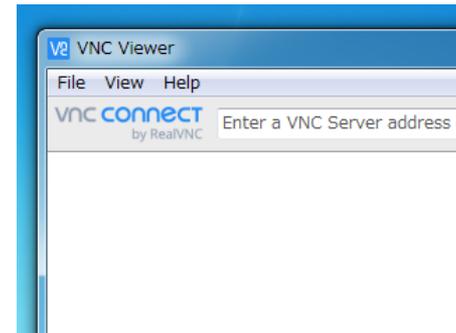
Raspbianの起動

「Finish」。 ショートカットアイコンが生成。最起動する



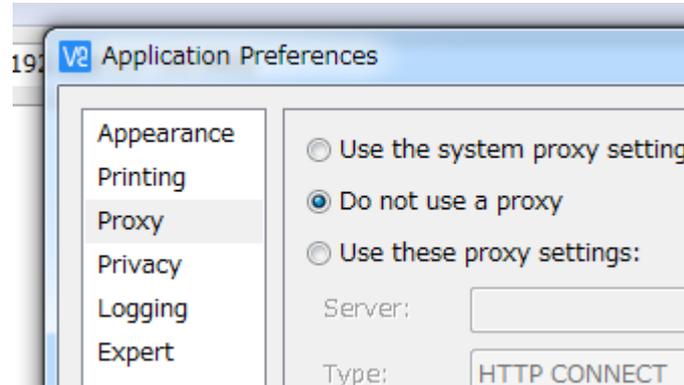
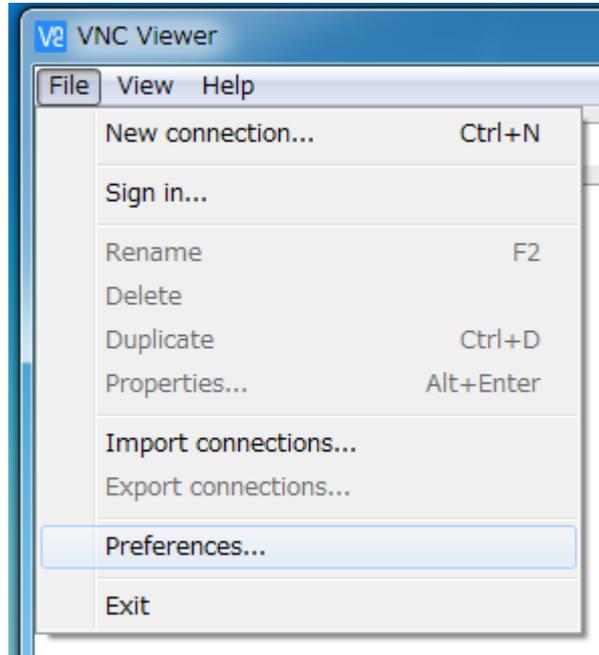
Raspbianの起動

VNCにチェックをいれてあるので WindowsPC にインストールした VNC Viewer を起動する。

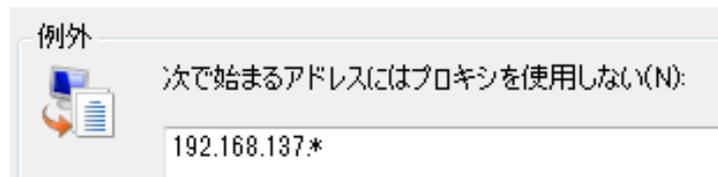


Raspbianの起動

プロキシ環境の場合、File→Preferences.. から Proxy → 「Do not use a proxy」を選択する。

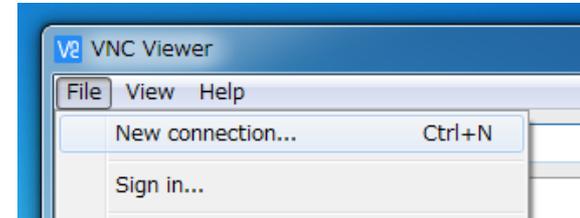


また、あらかじめPCで以下のようにproxyから除外している場合はデフォルト設定でもOK

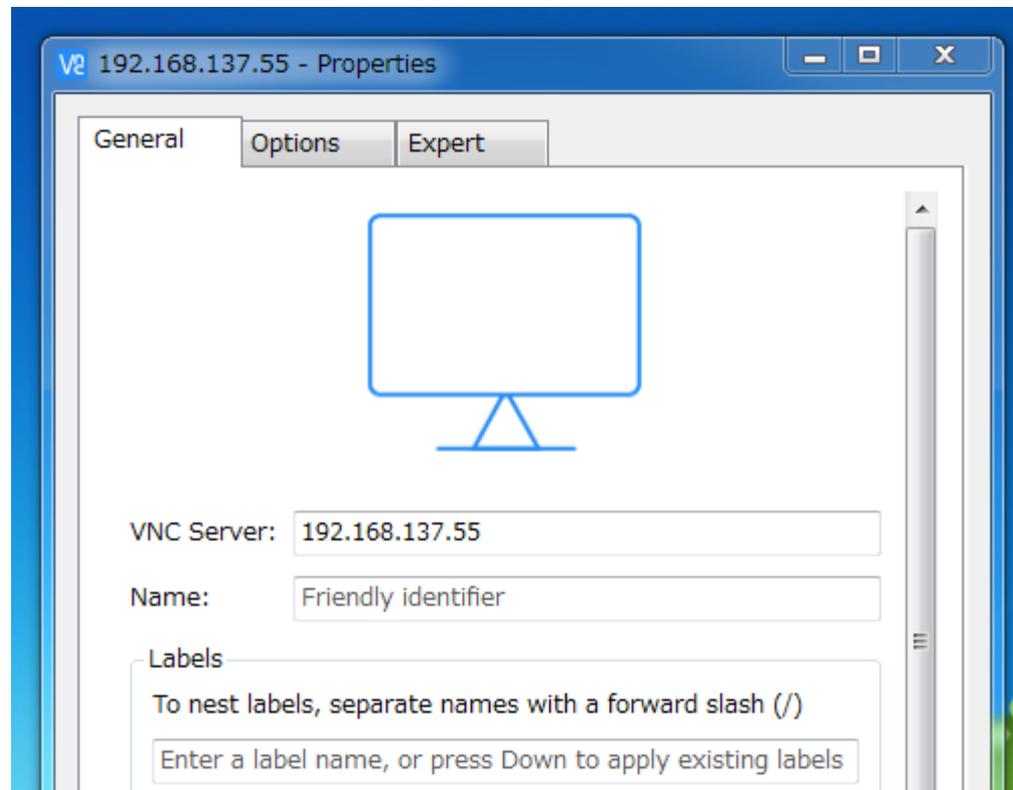


Raspbianの起動

VNC Viewerにて「File」→「New connection」。

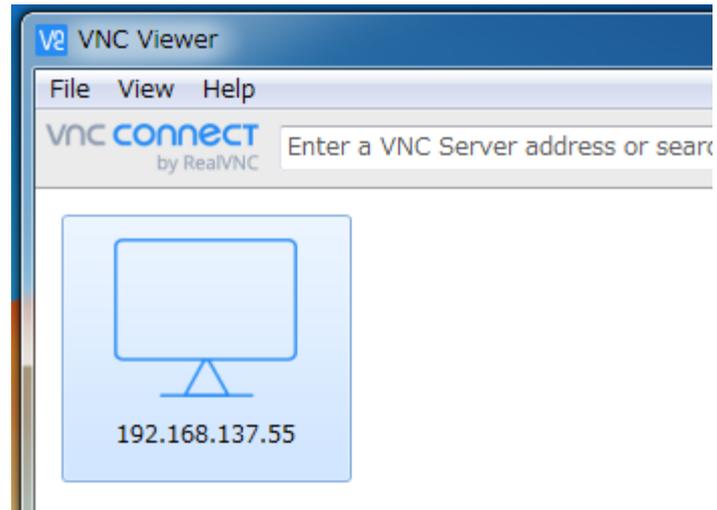


次回の接続の便利性的のためIPアドレスに対するサーバー名を登録できる



Raspbianの起動

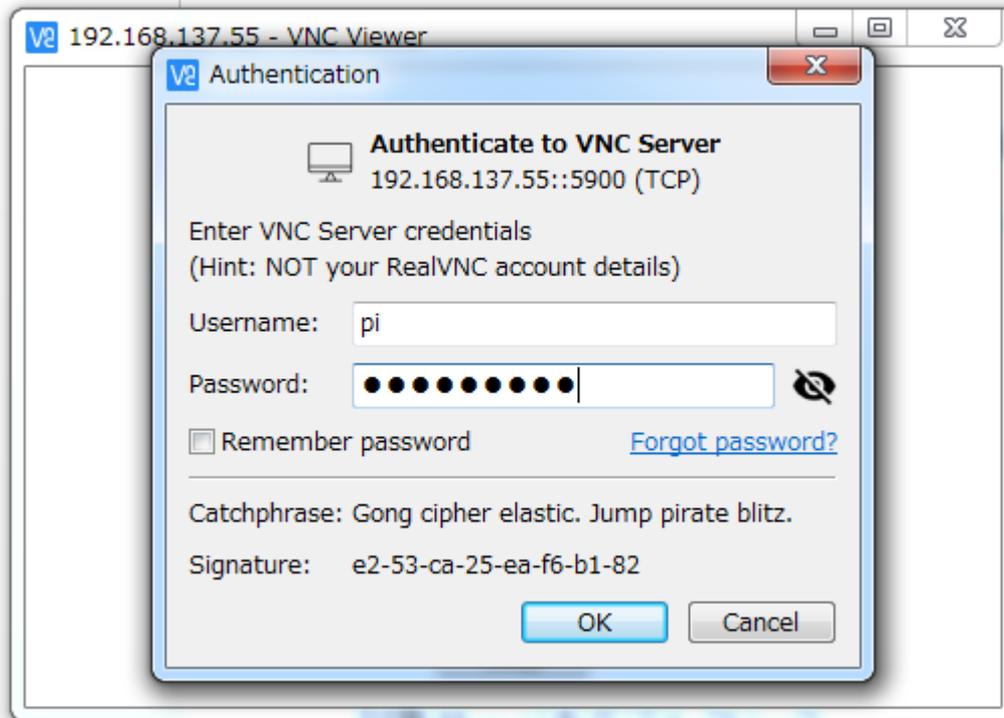
「sample」名にて登録画面が確認できる。



対象サーバをダブルクリック

Raspbianの起動

サーバのIPアドレスとポート番号(5900)が確認できる。
デフォルトは username「pi」、Password「raspberrry」

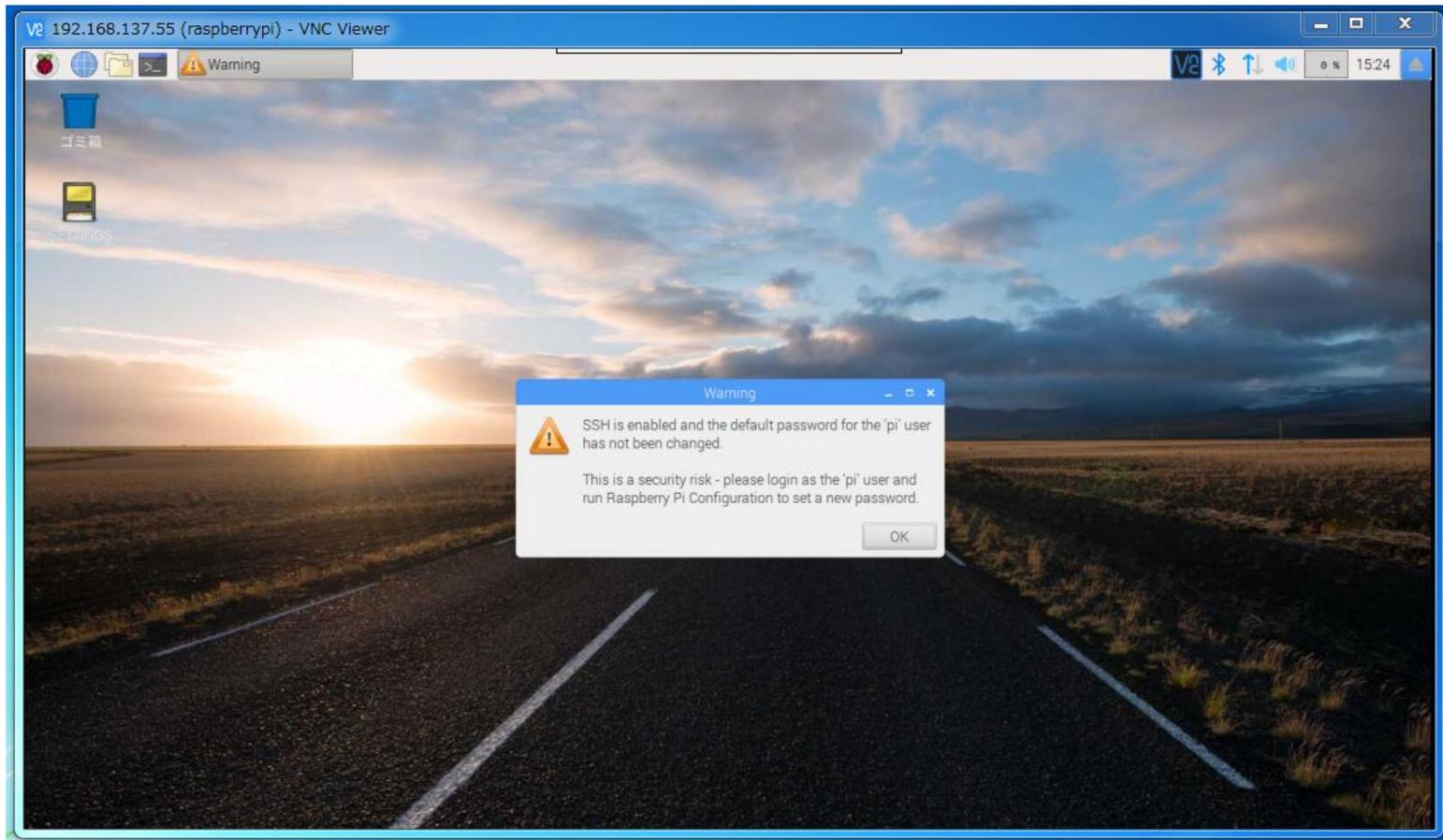


(なお、tightvncserverのポート番号は5901)

Raspbianの起動

Raspiウィンドウが現れる。

モニタ出力と同じ解像度で現れる。Skype画面も本モニタで確認できる。



Raspbianの起動

モニタ出力と同じ解像度で現れるのは
「設定」→「RasoberryPiの設定」 から「解像度の設定」が
「Default-preferred monitor settings」になっているため。

