

# 「付録2」

## Raspbian 時間設定

Raspbianのいずれかのバージョンからntpはサポートされなくなった。

ntpクライアントにsystemd-timesyncdが使われているのでntpクライアントやntpdate)と競合する。

インターネットに接続(ポートNo123が伝わる)ではそのままデフォルトでインターネット上のサーバと同期がとられるが、当施設のプロキシ環境では以下の状態となる。

なお、以下の内容は同期がとれない場合の対処であって、同期時間が図れる環境はスルーしてよい。(備忘録的情報)

```
$ sudo timedate status
```

```
pi@raspberrypi:~$ timedatectl status
Local time: 月 2019-04-08 20:48:13 JST
Universal time: 月 2019-04-08 11:48:13 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:~$
```

## 「付録2」 Raspbian 時間設定

ネット情報によれば、`/etc/systemd/timesyncd.conf` を編集する。  
バックアップをとって編集する。

```
$ sudo cp timesyncd.conf timesync.conf.org
```

```
pi@raspberrypi:~ $ cd /etc/systemd/  
pi@raspberrypi:/etc/systemd $ sudo cp timesyncd.conf timesyncd.conf.org
```

```
$ sudo nano timesyncd.conf
```

```
pi@raspberrypi:/etc/systemd $ sudo nano timesyncd.conf
```

以下のコメントをはずし、

NTP= 以降にntpサーバを登録する。ここでは172.16.0.1(LAN内のタイムサーバ)を登録する。

```
[Time]  
NTP=172.16.0.1  
#FallbackNTP=0.debian.pool.ntp.org 1.d
```

## 「付録2」 Raspbian 時間設定

systemd-timesyncdを再起動する。

```
pi@raspberrypi:/etc/systemd $ sudo systemctl restart systemd-timesyncd
```

同期履歴を調べる

```
$ systemctl status systemd-timesyncd(.service)
```

```
pi@raspberrypi:/etc/systemd $ sudo systemctl status systemd-timesyncd.service
```

```
pi@raspberrypi:/etc/systemd $ sudo systemctl status systemd-timesyncd.service
● systemd-timesyncd.service - Network Time Synchronization
   Loaded: loaded (/lib/systemd/system/systemd-timesyncd.service; enabled; ver
   Drop-In: /lib/systemd/system/systemd-timesyncd.service.d
            └─disable-with-time-daemon.conf
   Active: active (running) since Mon 2019-04-08 20:55:00 JST; 2 months 9 days
   Docs: man:systemd-timesyncd.service(8)
  Main PID: 1390 (systemd-timesyn)
   Status: "Synchronized to time server 172.16.0.1:123 (172.16.0.1)."
```

```
CGroup: /system.slice/systemd-timesyncd.service
         └─1390 /lib/systemd/systemd-timesyncd

4月 08 20:55:00 raspberrypi systemd[1]: Starting Network Time Synchronization
4月 08 20:55:00 raspberrypi systemd[1]: Started Network Time Synchronization.
6月 18 11:41:32 raspberrypi systemd-timesyncd[1390]: Synchronized to time ser
```

# 「付録2」 Raspbian 時間設定

同期できない場合、systemd.serviceの応用サンプルとしてntpdateを使用する方法も示す。ntpdateをインストールする。(timedatectlは停止させ、競合を避けること)

```
pi@raspberrypi:~ $ sudo apt-get install -y ntpdate
```

実行するスクリプトファイル(スクリプトファイルにするには理由がある。後述)を作成する。ここではcheckdate.shのファイルとする。ここでは/home/pi上に作成している。

```
pi@raspberrypi:~ $ nano checkdate.sh
```

ntpdateは/usr/sbinにインストールされており、300秒(5分)おきに同期をとる設定とする。ntpサーバは172.16.0.7とする。

```
GNU nano 2.7.4          ファイル: checkdate.sh

#!/bin/sh

while true
do
    sudo /usr/sbin/ntpdate 172.16.0.7
    sleep 300
done
```

# 「付録2」 Raspbian 時間設定

checkdate.shのパーミッションを変更する。

```
ファイル(F) 編集(E) タブ(T) ヘルプ(H)  
pi@raspberrypi:~ $ chmod 755 checkdate.sh
```

続いて、systemdのサービスファイルを作成する。

ここでは/etc/systemd/system に get\_date.serviceの名前で作成する。

```
pi@raspberrypi:~ $ cd /etc/systemd/system/  
pi@raspberrypi:/etc/systemd/system $ sudo nano get_date.service
```

```
GNU nano 2.7.4                ファイル: get_date.service  
  
[Unit]  
Description = Get_date Auto Start  
  
[Service]  
Type=simple  
ExecStart=/home/pi/checkdate.sh  
  
[Install]  
WantedBy=multi-user.target  
█
```

## 「付録2」 Raspbian 時間設定

スクリプトファイルのcheckdate.sh をフルパスで指定。

get\_dateをスタートさせる。

```
$ sudo systemctl enable get_date(.service)
```

```
$ sudo systemctl start get_date(.service)
```

```
pi@raspberrypi:/etc/systemd/system $ sudo systemctl enable get_date
pi@raspberrypi:/etc/systemd/system $ sudo systemctl start get_date
pi@raspberrypi:/etc/systemd/system $
```

しばらくすると同期に成功する。

## 「付録2」 Raspbian 時間設定

再起動後の自動立ち上がり後をチェックする。

```
$ sudo systemctl status get_date (.service)
```

最初の1回目はntpサーバーが探せず、同期処理に失敗している。

```
[*]: Started Get_date Auto Start.  
[*]: root : TTY=unknown ; PWD=/ ; USER=root ; COMMAND=/usr/sbin/ntpdate 172.16.0.7  
[*]: pam_unix(sudo:session): session opened for user root by (uid=0)  
e.sh[288]: 31 Aug 15:14:48 ntpdate[341]: no servers can be used, exiting  
[*]: pam_unix(sudo:session): session closed for user root  
[*]: root : TTY=unknown ; PWD=/ ; USER=root ; COMMAND=/usr/sbin/ntpdate 172.16.0.7  
[*]: pam_unix(sudo:session): session opened for user root by (uid=0)  
e.sh[288]: 31 Aug 15:20:19 ntpdate[891]: step time server 172.16.0.7 offset 24.275308
```

ループプログラムのおかげで

5分毎にntpdateの実行で同期が図られている。

スクリプトファイルにするのは最初の1回目のntpdateの同期処理は失敗しているので定期的な継続処理が必要となるため。

スクリプトファイルは簡単なプログラムが作成できるメリットがある。

## 「付録2」 Raspbian 時間設定

LAN内に適当なntpサーバがない場合、LAN内の開発支援PCであるubuntuPCをntpサーバになってもらうことが可能。

ubuntuPCにntpをインストールする。

```
user01@ubuntu:~$ sudo apt-get install -y ntp
```

/etc/ntp.confを編集する。

```
user01@ubuntu:~$ cd /etc
user01@ubuntu:/etc$ sudo cp ntp.conf ntp.conf.org
```

自マシンのハードウェア時間をもとにしたタイムサーバとするにはntp.confの文末に以下の2行を書き加えればよい。

stratum は stratum 10 にて設定を記述する。

IPアドレス 127.127.1.0 とは特別な番号アドレスである。

```
#broadcastclient

server 127.127.1.0
fudge 127.127.1.0 stratum 10
```

# 「付録2」 Raspbian 時間設定

ntpを再起動させる。

```
$sudo /etc/init.d/ntp restart
```

あるいは

```
$sudo service ntp restart
```

```
$ sudo systemctl restart ntp
```

```
user01@ubuntu:~$ sudo /etc/init.d/ntp restart  
[ ok ] Restarting ntp (via systemctl): ntp.service.  
user01@ubuntu:~$
```

\$ntpq -p にてサーバを検索する。

5分から10分経過後、LOCAL(0)に\*が付く。”st”欄 stratumが5で認識されている。

```
user01@ubuntu:~$ ntpq -p
```

remote	refid	st	t	when	poll	reach	del
0.ubuntu.pool.n	.POOL.	16	p	-	64	0	0.0
1.ubuntu.pool.n	.POOL.	16	p	-	64	0	0.0
2.ubuntu.pool.n	.POOL.	16	p	-	64	0	0.0
3.ubuntu.pool.n	.POOL.	16	p	-	64	0	0.0
ntp.ubuntu.com	.POOL.	16	p	-	64	0	0.0
*LOCAL(0)	.LOCL.	5	l	16	64	77	0.0

```
user01@ubuntu:~$
```

## 「付録2」 Raspbian 時間設定

開発支援のパソコンubuntuPCのntpサーバが192.168.137.3 の場合、ntpdateで確認できる。

```
pi@raspberrypi:~ $ sudo ntpdate 192.168.137.3
20 Jun 11:00:00 ntpdate[1216]: adjust time server 192.168.137.3 offset
pi@raspberrypi:~ $
```

# 「付録3」 VNCサーバの回復

起動アプリを確認する。

\$ ps a

```
pi@raspberrypi:~ $ ps -a
PID TTY          TIME CMD
 541 tty1          00:00:00 bash
18162 pts/0        00:00:00 Xtightvnc
18166 pts/0        00:00:00 xstartup
18169 pts/0        00:00:00 lxsession
18221 pts/0        00:00:00 openbox
```

VNCサーバが起動していれば停止する。ここでは PID=18162を停止する。

\$ kill 18162

```
pi@raspberrypi:~ $ kill 18162
pi@raspberrypi:~ $ ps -a
PID TTY          TIME CMD
 541 tty1          00:00:00 bash
18323 pts/0        00:00:00 ps
pi@raspberrypi:~ $
```

## 「付録3」 VNCサーバの回復

バージョンを重ねるうち tightvncserver (ポートNo.5901) と Raspi標準VNCサーバ (ポートNo.5900) の併用ができることが確認できた。

併用する場合はアンインストール処理は行わない。

(この部分の処理は飛ばす)

tightvncserver のアンインストール

```
$ sudo apt-get remove -y tightvncserver
```

```
pi@raspberrypi:~ $ sudo apt-get remove -y tightvncserver
```

/home/pi/.vnc を削除する。

```
$ sudo rm -rf /home/pi/.vnc
```

```
pi@raspberrypi:~ $ sudo rm -rf /home/pi/.vnc/
```

## 「付録3」 VNCサーバの回復

### 保存していたデータを元にもどす(1)

realvnc-vnc-server を /usr/share/doc フォルダにコピーする。

realvnc-vnc-server の退避場所が

/home/pi/VncRecallData/Usr\_share\_doc\_vnc\_back として

```
$ sudo cp -a /home/pi/VncRecallData/Usr_share_doc_vnc_back/realvnc-vnc-server /usr/share/doc/
```

```
pi@raspberrypi:~$ sudo cp -a /home/pi/VncRecallData/Usr_share_doc_vnc_back/realvnc-vnc-server /usr/share/doc
```

### 保存していたデータを元にもどす(2)

デフォルトの /usr/bin フォルダ内の vnc\* を書き戻す。

/home/pi/VncRecallData/Usr\_bin\_vnc\_back フォルダに退避しているとして

```
$ sudo cp -a /home/pi/VncRecallData/Usr_bin_vnc_back/* /usr/bin/
```

```
pi@raspberrypi:~$ sudo cp -a /home/pi/VncRecallData/Usr_bin_vnc_back/* /usr/bin
```

## 「付録3」 VNCサーバの回復

保存していたデータを元にもどす(3)

/usr/share/vnc フォルダを書き戻す。

/home/pi/VncRecallData/Usr\_share\_vnc\_back フォルダに退避しているとして

```
$ sudo cp -a /home/pi/VncRecallData/Usr_share_vnc_back/vnc
```

```
/usr/share/
```

```
pi@raspberrypi:~ $ sudo cp -a /home/pi/VncRecallData/Usr_share_vnc_back/vnc  
/usr/share
```

# 「付録3」 VNCサーバの回復

ちなみに併用状態の `vncserver` の実行ファイルは標準VNCが使用される。解像度の自由度が欲しい場合はリンク先を変える必要がある。

リンク先を `tightvncserver` のものに変えて異常なし。

併用時の `vncserver` のリンク先

```
15:10 vncpasswd -> /etc/alternatives/vncpasswd
2018 vncpipehelper
11:15 vncserver -> vncserver-virtual
2018 vncserver-virtual
2018 vncserver-virtuald
```

当初のRasPi標準VNCサーバのリンク先

```
11:31 vncserver -> vncserver-virtual
2018 vncserver-virtual
2018 vncserver-virtuald
2018 vncserver-x11
2018 vncserver-x11-core
2018 vncserver-x11-serviced
2018 vncserverui
```

## 「付録3」 VNCサーバの回復

リンク先をtightvncserverのものに変えてみる。

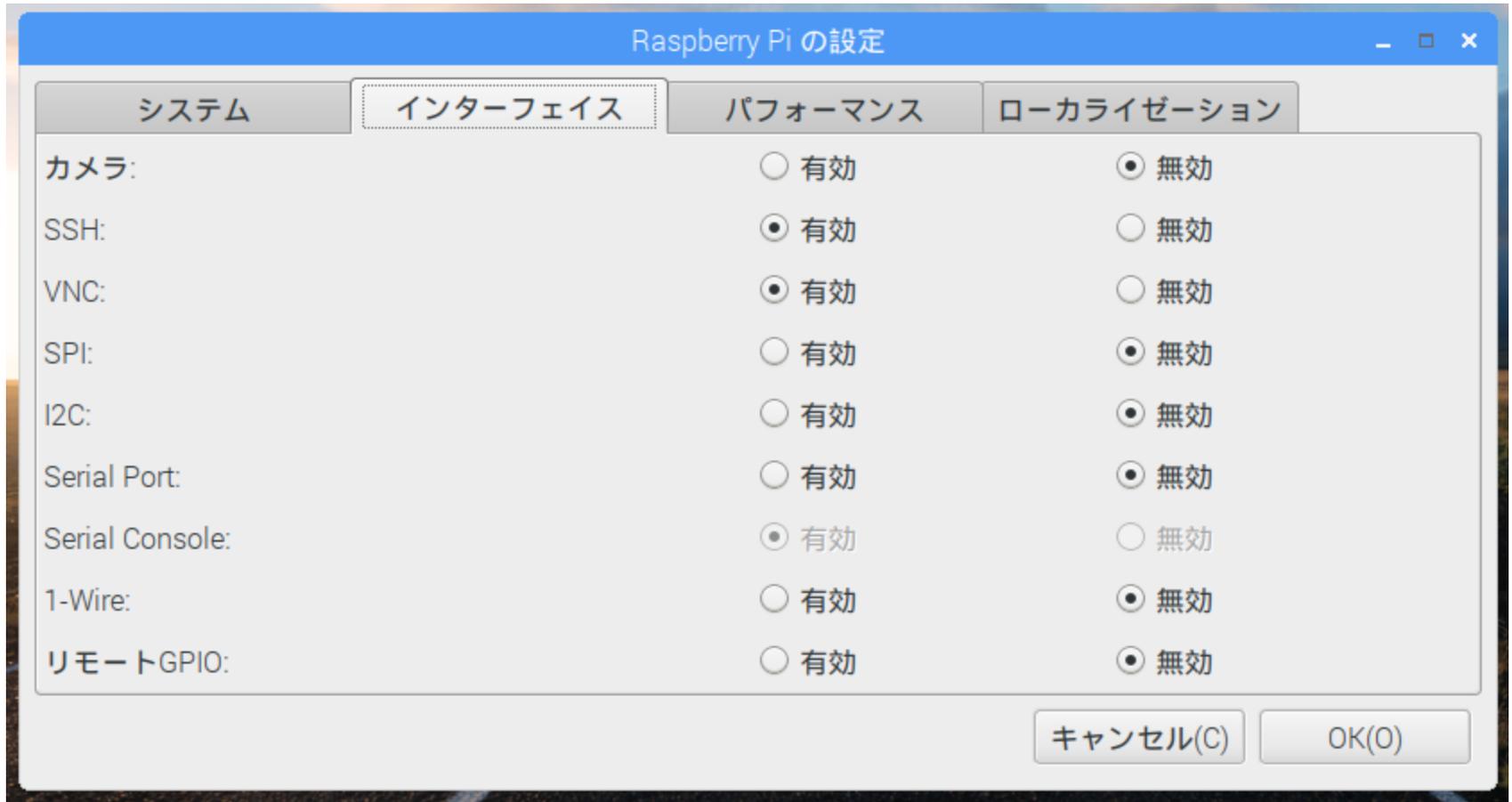
```
pi@raspberrypi:~/usr/bin $ sudo mv vncserver vncserver.bak
pi@raspberrypi:~/usr/bin $ sudo ln /etc/alternatives/vncserver vncserver
pi@raspberrypi:~/usr/bin $ ls -al vncserver*
lrwxrwxrwx 2 root root      23  6月 27 15:10 vncserver -> /usr/bin/tightvncserver
-rwxr-xr-x 1 root root 1038508  7月  2 2018 vncserver-virtual
-rwxr-xr-x 1 root root 2167488  7月  2 2018 vncserver-virtuald
```

ポート5900での受信待ちとなる。

```
pi@raspberrypi:~ $ netstat -lnt
稼働中のインターネット接続（サーバのみ）
Proto 受信-Q 送信-Q 内部アドレス          外部アドレス          状態
tcp    0      0 0.0.0.0:5900          0.0.0.0:*              LISTEN
tcp    0      0 0.0.0.0:22           0.0.0.0:*              LISTEN
tcp6   0      0 :::5900              :::*                   LISTEN
tcp6   0      0 :::22                :::*                   LISTEN
pi@raspberrypi:~ $
```

特に異常は現れず。デフォルトのサイズが選択される。

# 「付録3」 VNCサーバの回復



## 「付録3」 VNCサーバの回復

ただし、再々度 `tightvncserver` をインストールする場合は  
`/usr/bin/vncpasswd` が使えないとのメッセージが出て再インストールできない。  
`/usr/bin/vncpasswd` ファイルを削除してからインストールする。

```
$ sudo rm /usr/bin/vncpasswd
```

```
$ sudo apt-get install -y tightvncserver
```

```
pi@raspberrypi:~ $ sudo rm /usr/bin/vncpasswd  
pi@raspberrypi:~ $ sudo apt-get install -y tightvncserver
```

## 「付録4」 アナログモニタでの起動

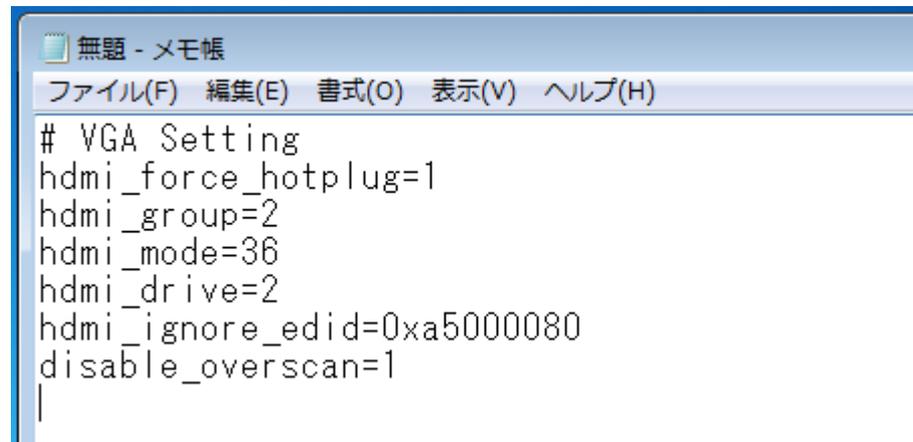
SDカードをRaspberryPiに取り付け、デジタルーアナログ変換機を介してアナログモニターに接続する。



## 「付録4」 アナログモニタでの起動

NOOBSはネットから使用できるOSを探し出してメニューとして仕上げてくれるが、Proxy環境ではRaspbian (2.8.2では他1点)でしか選択できないので他のOSを試す場合はダウンロードサイトに戻って個別にOSをダウンロードしてインストールする。

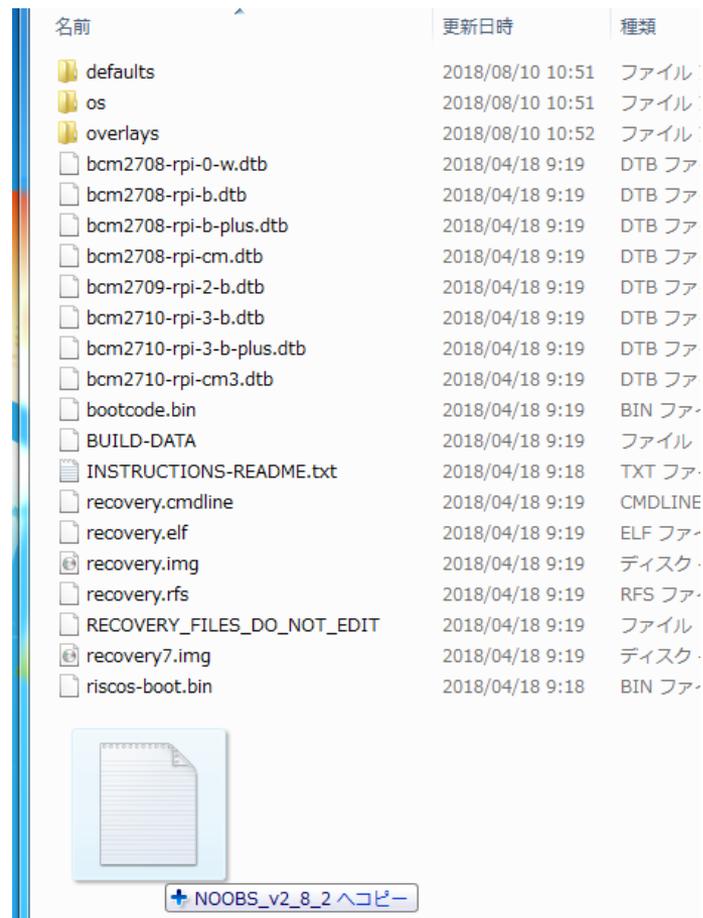
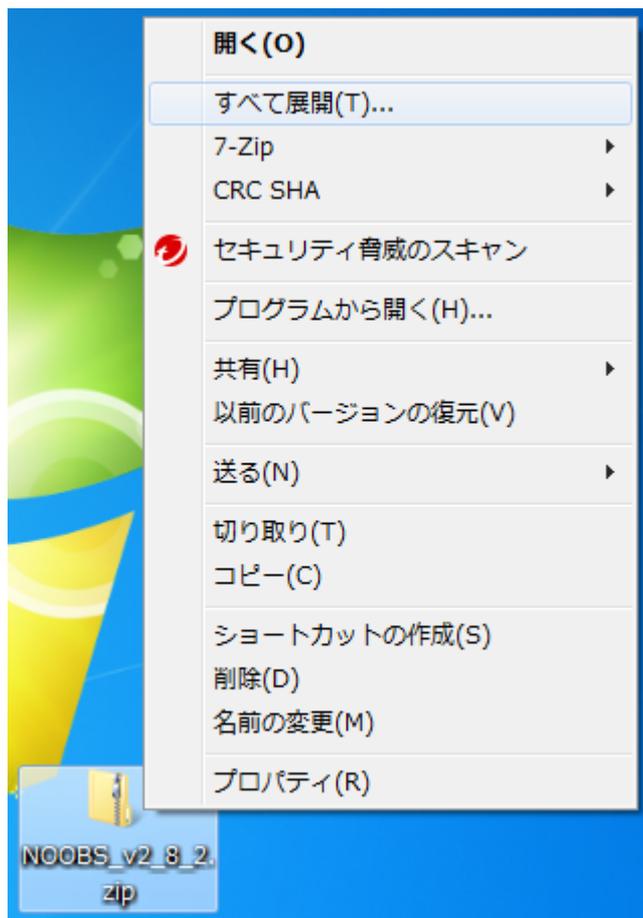
余剰のアナログモニタを使用するには以下内容のテキストファイルを作成する。ここではエディタとしてメモ帳を起動し、以下のテキスト文を入力し、config.txtの名前で保存する。



```
無題 - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
# VGA Setting
hdmi_force_hotplug=1
hdmi_group=2
hdmi_mode=36
hdmi_drive=2
hdmi_ignore_edid=0xa5000080
disable_overscan=1
|
```

# 「付録4」 アナログモニタでの起動

アナログモニタ使用時はNOOBSのzip 展開フォルダにconfig.txtを入れる。



## 「付録4」 アナログモニタでの起動

アナログモニタの機種によっては再起動後何も映らず、ブラックアウトする場合もあり、このときはこのあとに示すようなSDカードをPCに認識させ /boot/config.txt を編集する。

以下のような画面に初期設定画面が出ればそのままOK。映るには映るが下記画面のような正常な形であればRaspi上で編集も可能。



# 「付録4」 アナログモニタでの起動

ブラックアウトしまう場合はSDカードをPCで認識させて書き換える他ない。ここでは相性のよいubuntuマシンに認識させる。マイクロSDカードをSDカードアダプタのUSB機器を使ってPCに認識させる。

ここでは/media/user01に /SETTINGS , /boot , /root \* として自動認識される。(通常は/rootとなるが、ubuntuマシンの状態によっては/root0, /root1といった数字が付加される場合がある)



```
user01@ubuntu:/media/user01$ ls
SETTINGS boot root
user01@ubuntu:/media/user01$
```

## 「付録4」 アナログモニタでの起動

各フォルダのファイルシステムは `$ df -T` で確認できる。

```
user01@ubuntu:/media/user01$ df -T
Filesystem      Type      1K-blocks    Used Available Use% Mounted on
udev            devtmpfs  463172        0    463172   0% /dev
tmpfs           tmpfs     98516         1744 96772   2% /run
/dev/sda1       ext4      51341792    7108124 41595948 15% /
tmpfs           tmpfs     492568        0    492568   0% /dev/shm
tmpfs           tmpfs     5120          4     5116   1% /run/lock
tmpfs           tmpfs     492568        0    492568   0% /sys/fs/cgroup
tmpfs           tmpfs     98512         16    98496   1% /run/user/120
tmpfs           tmpfs     98512         28    98484   1% /run/user/1000
/dev/sdb7       ext4      13457240    4217976 8532628 34% /media/user01/root
/dev/sdb5       ext4      30701        398    28010   2% /media/user01/SETTINGS
/dev/sdb6       vfat      69553        22192 47362  32% /media/user01/boot
/dev/sdb8       ext4      99704        99704 0 100% /media/user01/14106
```

# 「付録4」 アナログモニタでの起動

/SETTINGSには3つの.confファイル

```
user01@ubuntu:/media/user01/SETTINGS$ ls
installed_os.json  lost+found  noobs.conf  wpa_supplicant.conf
```

下記の/bootには Raspbianの/bootに組み込まれるファイル群

```
user01@ubuntu:/media/user01/boot$ ls
COPYING.linux          bcm2709-rpi-2-b.dtb      fixup.dat              os_config.json
LICENCE.broadcom      bcm2710-rpi-3-b-plus.dtb  fixup_cd.dat          overlays
LICENSE.oracle        bcm2710-rpi-3-b.dtb     fixup_db.dat          start.elf
bcm2708-rpi-0-w.dtb   bcm2710-rpi-cm3.dtb     fixup_x.dat           start_cd.elf
bcm2708-rpi-b-plus.dtb  bootcode.bin            issue.txt              start_db.elf
bcm2708-rpi-b.dtb     cmdline.txt              kernel.img             start_x.elf
bcm2708-rpi-cm.dtb    config.txt                kernel7.img
```

```
user01@ubuntu:/media/user01/boot$
```

アナログモニタに関するコードはconfig.txtファイルに書き加える。

下記の/rootは、Raspbianのルートフォルダに相当する。

```
user01@ubuntu:/media/user01/root$ ls
bin  debootstrap  etc  lib          media  opt  root  sbin  sys  usr
boot dev          home lost+found  mnt    proc run  srv  tmp  var
user01@ubuntu:/media/user01/root$
```

## 「付録4」 アナログモニタでの起動

一方、ブラックアウトせず異状になりながらも画面操作ができるようなら  
/boot/config.txtをそのまま編集し、下記コードをデフォルト文以降に書き加える。

```
# VGA Setting
hdmi_force_hotplug=1
hdmi_group=2
hdmi_mode=36
hdmi_drive=2
hdmi_ignore_edid=0xa5000080
disable_overscan=1
```

または、df コマンドでSDカードは/dev/mmcblk0p1と認識しているので、  
/dev/mmcblk0p1に最初のconfig.txtファイルがあるのでエディタなどで対象部分の  
コードをコピーして書き加える。

```
pi@raspberrypi:~ $ df
ファイルシステム 1K-ブロック 使用 使用可 使用% マウント位置
/dev/root          13457248 4762768 7987844   38% /
/dev/mmcblk0p6    69553      22498  47055   33% /boot
tmpfs              94944         0  94944    0% /run/user/1000
/dev/mmcblk0p5    30701         398  28010    2% /media/pi/SETTINGS
pi@raspberrypi:~ $ █
```

```
pi@raspberrypi:~ $ sudo mount /dev/mmcblk0p1 /mnt
pi@raspberrypi:~ $ nano /mnt/config.txt █
```