

自動運転車両

評価車両模型 OS セットアップ

Saoto-Tech

[履歴]

Rev. 1.0 : 2018/4/27 初版 (土屋)

Rev. 2.0 : 2020/03/22 改訂 (土屋)

目次

1. はじめに.....	1
2. 制御システム.....	1
3. ラジコン.....	2
4. PC.....	2
5. Raspberry Pi.....	4
引用文献.....	8

1. はじめに

トレーラーラジコン模型ベースの自動運転評価車両について、
搭載した制御システムのセットアップ及び基本的な使い方を示す。

この取説の書き方について。

PC のコマンドライン

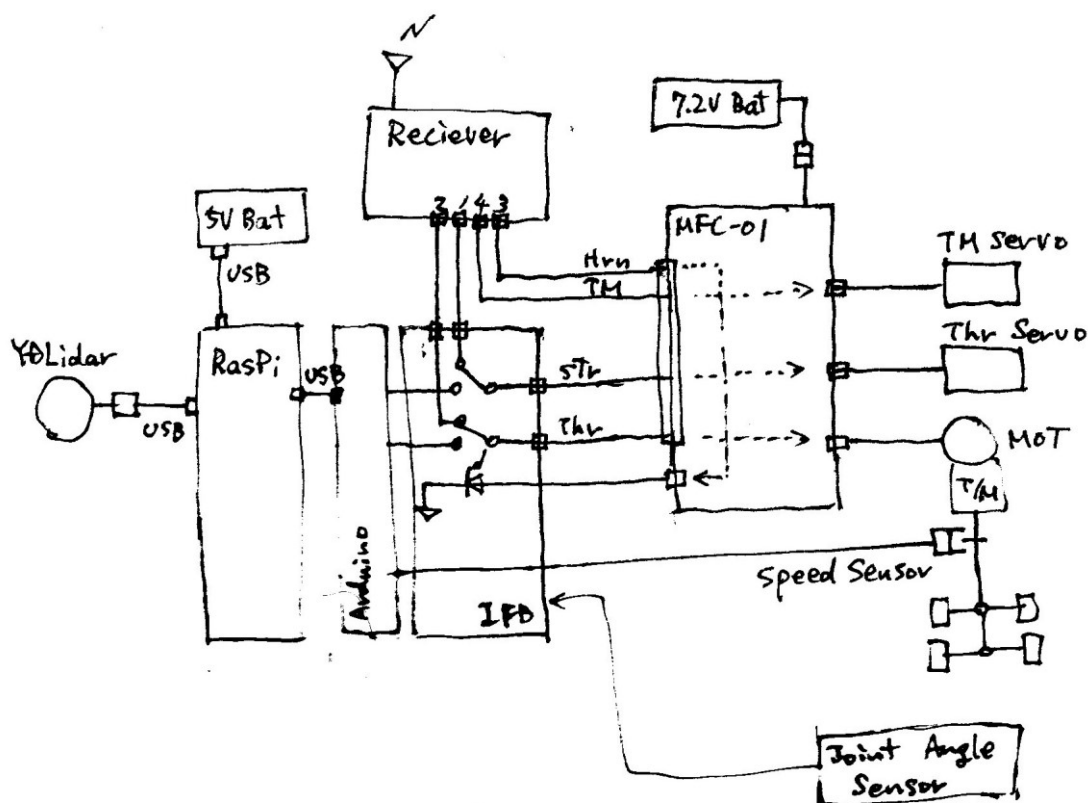
```
saoto@snote17:~$
```

raspi でのコマンドライン

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

2. 制御システム

- ・ 車両のメイン制御装置には Raspberry Pi (RasPi)
- ・ ラジコン操作も生かすため、ラジコン受信機からの指令信号と RasPi からの指令信号を、フォト CMOS リレーで切り替える。フェールセーフ上、デフォルトは受信機側。
- ・ フォグランプの ON/OFF に対応してラジコン操作と自動運転を切り替える。
- ・ 入出力 (デジタル・アナログ) インターフェースとして Arduino を設置。



3. ラジコン

デフォルトは、ラジオコントロール状態。

フォグランプを ON にするとフォト MOS リレーが切り替わり、ステアリングとモータのコントロールは受信機から分離され Arduino からの信号が伝達される。

フォグランプを ON/OFF には：

- ・ ラジコン送信機の右スティックの下操作(3ch)による（トリムを下げる必要あり。詳細はタミヤのコントロールユニット操作説明参照）
- ・ 車両側のコントロールユニットで手動で切り替えることも可能。しかし、送信機が ON でないとモータ・ステアサーボは駆動されない。

モータが駆動されていると、送信機からのフォグランプ ON/OFF が無視される。従って、走行中の自動運転の強制停止には、送信機の電源を切る。（モータ駆動するためには、受信機から MFC へミッションサーボ指令信号が入らないといけならしい。送信機を切ると、この信号も止まるので、モータ駆動も止まる）

起動直後は、駆動できるまで少し待つ必要あり

前進から後進にレバーを倒しても、1 回目は駆動禁止。（Arduino プログラムでは対応済み）

4. PC

RasPi と ROS で連携するために、PC 側も OS は ubuntu の必要あり。

ubuntu が PC に対応しているかは、次のサイトで確認。

<https://certification.ubuntu.com/desktop>

2020 年度仕様：

PC：DELL Vostro 14 5000(5490) COREi7

OS：ubuntu 18.04（デュアルブートで Windows10 も残す）

ROS：Melodic

使う必要はないですが、Windows は、ID: test、Password: test で登録してあります。

ubuntu も、ID: test、Password: test で登録してあります。

BIOS の設定は、電源オン直後に F2 キー。

ubuntu 18.04 インストール

ここからダウンロード。

<https://jp.ubuntu.com/download>

下記のサイトを参考にインストール。

<https://www.dell.com/support/article/ja-jp/sln301754/dell-pc-%E3%81%AB%E3%83%87%E3%83%A5%E3%82%A2%E3%83%AB%E3%83%96%E3%83%BC%E3%83%88%E3%81%A8%E3%81%97%E3%81%A6-ubuntu-%E3%81%8A%E3%82%88%E3%81%B3-windows-8-%E3%81%BE%E3%81%9F%E3%81%AF10%E3%82%92%E3%82%A4%E3%83%B3%E3%82%B9%E3%83%88%E3%83%BC%E3%83%AB%E3%81%99%E3%82%8B%E6%96%B9%E6%B3%95?lang=ja>

<https://qiita.com/medalotte/items/4bb5cfa709e93d044f1c>

<https://qiita.com/medalotte/items/4bb5cfa709e93d044f1c>

インストール前に、Bios 設定で、Secure Boot を OFF にする。

500GB SSD の内、約 250GB を ubuntu 用に開放。

PCIe M2 の SSD が認識されず、BIOS で SATA 設定を AHCI に変更する。

<https://ja.stackoverflow.com/questions/49003/linux%E3%81%AE%E3%82%A4%E3%83%B3%E3%82%B9%E3%83%88%E3%83%BC%E3%83%AB%E6%99%82%E3%81%ABm-2-ssd%E3%81%8C%E8%AA%8D%E8%AD%98%E3%81%95%E3%82%8C%E3%81%AA%E3%81%84>

インストール後に wifi が使えず

最新カーネル（linux-image-5.3.0-42-generic）だとだめだが、古いカーネル

（linux-image-5.3.0-28-generic）で起動すれば wifi が使えるので、最新カーネルはアンインストールする。

<https://unix.stackexchange.com/questions/432393/downgrade-linux-kernel-without-grub>

上記のように、ubuntu インストールでは、往々にして問題が生じます。その都度、ネットで調べながら解決していきます。

ROS melodic をインストール

ubuntu 18 に対応する ROS melodic をインストール。

<http://wiki.ros.org/melodic/Installation/Ubuntu>

virtual router のインストール

create_ap をインストールして、PC をルータとして raspi を接続。

<https://qiita.com/KuwabataK/items/5903c7584657151d576a>

こんな感じに起動

```
saoto@snote17:~$ sudo create_ap wlp2s0 wlp2s0 <SSID> <passphrase>
```

プリインストールでは、<SSID>を note、<passphrase>を note2020 と設定している。

このコマンドは、virtual_router というに名前の file に保存して、chmod で実行可能にしておく。

wifi の注意事項

PC の wifi ハードウェアによるようだが、create_ap を起動するより前に 5GHz のアクセスポイントに接続していると、create_ap が起動できないことあり。

その場合は、いったんアクセスポイントとの接続を切断し、create_ap を起動してから再度アクセスポイントに接続すれば良い。

2.4GHz のアクセスポイントであれば問題ない。

internet に関して

インターネット接続の無い状態で ROS をネットワークで分散処理させると、ipv6 の関係でうまくいかないらしい。

ipv6 の disable で解消する

<https://answers.ros.org/question/218739/connecting-to-a-ros-master-without-internet/>

<https://answers.ros.org/question/269182/correct-ros-master-uri-format-for-ipv6/>

5. Raspberry Pi

「RaspberryPi で学ぶ ROS ロボット入門」 [上田隆一, 2017]を参考にセットアップ。

仕様：

Raspberry Pi 3 Model B v1.2

OS: ubuntu Server (GUI 無し) 16.04 LTS(long-term support)

ROS: kinetic

SD カード：micro 16GB

ubuntu と ROS のインストール

参考文献サイトより。

https://github.com/ryuichiueda/raspimouse_book_info

os_images.md を参照

Ubuntu 16.04 Server の「自動アップデートを止めた Raspberry Pi 3 用～」がうまくいった。

(他は wifi に問題発生)

これならば、ROS もインストール済み。

microSD を PC に挿して /dev のファイルを確認。

土屋は以下のファイルだった。

```
saoto@snote17:~$ ls -l /dev/sd*  
brw-rw---- 1 root disk 8, 0  4月 26 14:59 /dev/sda  
brw-rw---- 1 root disk 8, 1  4月 26 14:59 /dev/sda1
```

/dev/mmcblk0* の場合もあるらしい。

ファイルシステムの確認

```
saoto@snote17:~$ df -h  
Filesystem      Size  Used Avail Use% Mounted on  
(中略)  
/dev/sda1       15G   32K   15G   1% /media/saoto/3935-3331
```

sda1 が /media~ にマウントされていたので、アンマウントする

```
saoto@snote17:~$ sudo umount /dev/sda1
```

microSD にダウンロードしたイメージを書き込む。

(time は時間を測るために使っているだけ)

```
saoto@snote17:~$ time xzcat  
ubuntu-16.04-preinstalled-server-armhf+raspi3-ros-noupgrade-rtmouse-catkin.img.xz  
| sudo dd bs=1MB of=/dev/sda
```

10 分ほどかかる。

microSD を RasPi に挿し microUSB 端子から給電する。

初回のみモニタとキーボードと LAN を接続。

(PC から RasPi の IP アドレスを探索すれば、初回からモニタなしでもできる)

(LAN を接続しないと、立ち上げに 5 分待つ)

login/password は、ubuntu/ubuntu

IP アドレスチェック

```
ubuntu@ubuntu:~$ ip addr show eth0
```

ここからは、PC から遠隔操作する。

(そのまま直接使う場合は、日本語表示や日本語キーボードの設定を直す必要がある)

```
saoto@snote17:~$ ssh ubuntu@192.168.x.x # アドレスチェックで得た IP アドレス  
(略)
```

```
Are you sure you want to continue connecting (yes/no)? yes ←yesと入力
Warning: Permanently added '192.168.12.110' (ECDSA) to the list of known hosts.
ubuntu@192.168.12.110's password:
Welcome to Ubuntu 16.04 LTS (GNU/Linux 4.4.0-1009-raspi2 armv7l)

* Documentation:  https://help.ubuntu.com/
(略)
ubuntu@ubuntu:~$
```

これで PC から raspi に入ったので、今後はモニターとキーボードは必要ない。

下記にあるシェルスクリプトを実行。自動アップデートされる前にすぐにやる。

https://github.com/ryuichiueda/raspimouse_book_ubuntu_init/blob/master/after_os_install.bash

スクリプトの内容を以下に転記する。これをファイルとして保存して実行してもよい。

```
#!/bin/bash
# (c) 2017 Ryuichi Ueda
# This software is released under the MIT License, see LICENSE.

# usage: This script stabilizes Ubuntu server 16.04 for Raspberry Pi 3
# at https://wiki.ubuntu.com/ARM/RaspberryPi. This script must be used
# just after the installation as soon as possible.

tmp=/tmp/$$

### purge of the cloud-init ###
sudo apt -y purge cloud-init

### remove the bug on the device tree address ###
cat /boot/firmware/config.txt |
sudo tee /boot/firmware/config.txt.org |
sed 's/device_tree_address=0x100/device_tree_address=0x02008000/' |
sed 's/device_tree_end=0x8000/#&/' |
sudo tee $tmp-config

sudo mv $tmp-config /boot/firmware/config.txt

### stop network device update ###
echo linux-firmware-raspi2 hold |
sudo dpkg --set-selections

### update ###
```

```
sudo apt update
sudo apt -y upgrade

### install WiFi tools ###
sudo apt -y install wireless-tools wpasupplicant

sudo reboot
```

スクリプトに十分以上かかります。下記のことをやってくれます。

- cloud-init の追い出し
- バグフィックス（これをしないと自動アップデートで立ち上がらなくなる）
- ネットワークデバイスのアップデート停止（これをしないと自動アップデートで wifi が使えなくなる）
- アップデートとツールインストール
- リブート

リブートしたら、ssh でログインし直す。

ここから wifi 接続

ネットワーク設定ファイルの変更

```
ubuntu@ubuntu:~$ sudo vi /etc/network/interfaces
有線の設定をコメントアウトして使えなくする
#source /etc/network/interfaces.d/*.cfg

次の4行を追加
auto wlan0
iface wlan0 inet dhcp
wpa-conf /etc/wpa_supplicant/virwifi.conf      ファイル名 virwifi は任意
wireless-power off
```

wifi 設定ファイル作成。

```
ubuntu@ubuntu:~$ sudo -s
root@ubuntu:~# wpa_passphrase SSID 暗号化キー > /etc/wpa_supplicant/virwifi.conf
root@ubuntu:~# cat /etc/wpa_supplicant/virwifi.conf      内容確認
```

wifi 立ち上げ。

```
root@ubuntu:~# ifup wlan0
Internet Systems Consortium DHCP Client 4.3.3
Copyright 2004-2015 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/
```

```
Listening on LPF/wlan0/b8:27:eb:79:b2:17
Sending on   LPF/wlan0/b8:27:eb:79:b2:17
Sending on   Socket/fallback
DHCPDISCOVER on wlan0 to 255.255.255.255 port 67 interval 3 (xid=0x56d5fa51)
DHCPDISCOVER on wlan0 to 255.255.255.255 port 67 interval 8 (xid=0x56d5fa51)
DHCPREQUEST of 192.168.12.110 on wlan0 to 255.255.255.255 port 67
(xid=0x51fad556)
DHCPOFFER of 192.168.12.110 from 192.168.12.1
DHCPACK of 192.168.12.110 from 192.168.12.1
bound to 192.168.12.110 -- renewal in 38162 seconds.
```

RasPi に 192.168.12.110 が割り振られて、wifi 接続できました。
次回からは、このアドレスで ssh 接続する。

reboot して、LAN を外し、wifi 経由で ssh 接続できるか確認する。

```
root@ubuntu:~# reboot
```

wifi 接続がうまくいかない場合、interfaces 設定で LAN も止めているので、またモニターとキーボードが必要になります。

参考文献の付録 C に以下の方法等あり

- SD カードの容量を全部使えるようにする
- 時刻合わせ
- sudo の warning をなくす

引用文献

小倉 崇. (2015). ROS ではじめるロボットプログラミング. 工学社.
上田隆一. (2017). RaspberryPi で学ぶ ROS ロボット入門. 日経 BP.