

チャレンジサイト・メカニックカモノハシ 2019

Ubuntu・Vim の基礎知識 マイクロマウスシミュレータ導入

ER17045 立道壱太郎

2019 年 8 月 19 日

1 この資料について

メカニックカモノハシではマイクロマウス大会に参加することでメンバーの技術向上を図ります。資料等は github に公開しているので、適時ダウンロードしてください。

```
$ git clone https://github.com/platypus5384/micro_mouse.git
```

また、PC・Ubuntu の基本操作、vim の基本操作、プログラミング言語の基礎知識 (if 文・for 文・配列等) が多少 (講義を受けた、1 週間くらい使ったことがある程度) を想定しています。最初は復習から始めますが、日常生活でも練習しておいてください。

2 動作環境について

Ubuntu16.04LTS

ROS Kinetic

での動作を前提としています。Ubuntu18.04LTS 等でも問題ないと思いますが、ros 関連のパッケージ名が違ってくるので注意 (kinetic を melodic 等に変えるだけなので簡単)

3 Terminal 及び Vim の基礎知識

Windows は、ほとんどの作業をマウスで操作する事が出来ます(GUI,GraphicalUserInterface)、Ubuntu でもマウス操作をすることが出来ますが、基本的にはコマンド入力による操作(CUI,CharacterUserInterface 又は CLI,CommandLineInterface) が主になります。慣れてくると、CUIの方が作業効率が良くなる(自由度が高く、script による自動化も簡単に出来る)ので、これを機会に慣れましょう。

このセクションでは、Terminal 及び Vim を使うにあたって作業効率が格段に上がる基礎知識を紹介するので100%覚えましょう。

3.1 Terminal

デフォルトの Terminal を想定しています。Terminator 等では通用しないかもしれません。

3.1.1 Terminal の開き方

一応言っておくと、Ctrl と Alt はキーボード左下にあります。T は Shift キーを押さずにです。

```
Ctrl + Alt + T
```

3.1.2 Terminal 内でのタブの開き方

```
Ctrl + Shift + T
```

3.1.3 Terminal 内でのタブ間の移動

```
Alt + [移動したいタブの左から数えた順番]
```

3.1.4 Terminal の文字の大きさを変更する

```
Ctrl + Shift + 「+」  
Ctrl + 「-」
```

3.1.5 コマンド実行中に強制中断

強制中断出来ない時もあります。

```
Ctrl + C
```

3.1.6 Terminal タブを閉じる

タブが一個しかなければウィンドウを閉じます。

```
Ctrl + D
```

3.2 Vim

Vim とはエディタです。Vim の強みは CUI 上で動くこと、自由度の高いコマンド操作です。

CUI 上で作業できるのが強みという点は、のちのち理解できると思います。

Vim ではコマンドを打つことで、コピペ、文字列置換やカーソル移動などが行えます。また、Vim のコマンドはプログラムのような物で、自由度が高く使いこなせば作業効率が凄いことになるらしい。プログラムを書く際には正直 Atom や VisualStudio の方が便利ですが、少し書く分には非常に便利で、また、どの PC にも基本的に入っているものなので、是非 Vim に慣れましょう。

以下のサイトにわかりやすく載っていますが、量が多いので最低限これだけは、というのを紹介します。使いながら覚えましょう。

```
https://qiita.com/FBH9999/items/a8ec99bed08592a9d69a
```

3.2.1 Vim のインストール

Vi というのはデフォルトで入っていますが、Vim はデフォルトで入っていません。ターミナル上で以下を実行してください。（\$を入力する必要はありません。\$から始まる行はターミナル上でのという意味です。）

```
$ sudo apt install vim
```

3.2.2 Vim の起動

```
$ vim [ファイル名]
$ vim test.py
$ vim (チュートリアルを開きます)
```

3.2.3 編集モード

編集モード（文章が書ける）に移行します。

```
a (そのカーソル後から編集モードに)
i (そのカーソルから編集モードに)
```

3.2.4 コマンドモード

コマンドモードに移行します。

```
Esc
又は、
Alt + コマンドの一文時目
```

3.2.5 コマンド

```
:w (上書き保存)
:q (終了)
:!q (強制終了 (保存せずに終了))
:wq (保存して終了)
```

```
:set number 行番号表示  
:[数字] その行に移動
```

以下のコマンドは [数字] + コマンド とすることで一気にn 回、n 行できます。

```
h j k l それぞれ、左、下、上、右に移動( 矢印でも移動できる )  
u (1回作業を戻る (undo))  
Ctrl + r (1回作業を進める (redo))  
yy (1行コピー)  
dd (1行コピー & 削除)  
p (ペースト)
```

4 マイクロマウスパッケージの導入

ROS が入っている、環境設定等が終わっている、というのを前提とします。(roscore 出来れば問題なし)

4.1 workspace の作成

メカニックカモノハシ用の workspace を新たに作りましょう。

```
$ mkdir -p ~/mp_ws/src  
$ cd ~/mp_ws/src/  
$ catkin_init_workspace  
$ echo "source ~/mp_ws/devel/setup.bash" >> ~/.bashrc
```

4.2 マイクロマウスパッケージの導入

github から、パッケージを clone して catkin_make します。

```
$ cd ~/mp_ws/src  
$ git clone https://github.com/platypus5384/micro_mouse.git  
$ cd ~/mp_ws  
$ catkin_ws  
$ source ~/.bashrc
```

以下のコマンドを入力し、ディレクトリを移動できれば成功です。

```
$ roscd micro_mouse
```

5 関連パッケージの導入

マイクロマウスパッケージの動作に必要な関連パッケージを導入します。以下のコマンドを実行してください。(全部で一行)

ソースコード 1 roscd

```
sudo apt install ros-kinetic-turtlebot ros-kinetic-turtlebot-msgs ros-kinetic-turtlebot-teleop
```

6 micro_mouse 動作テスト

付いていると思いますが、一応、ファイルに権限を付けます。以下を実行してください。

ソースコード 2 roscd

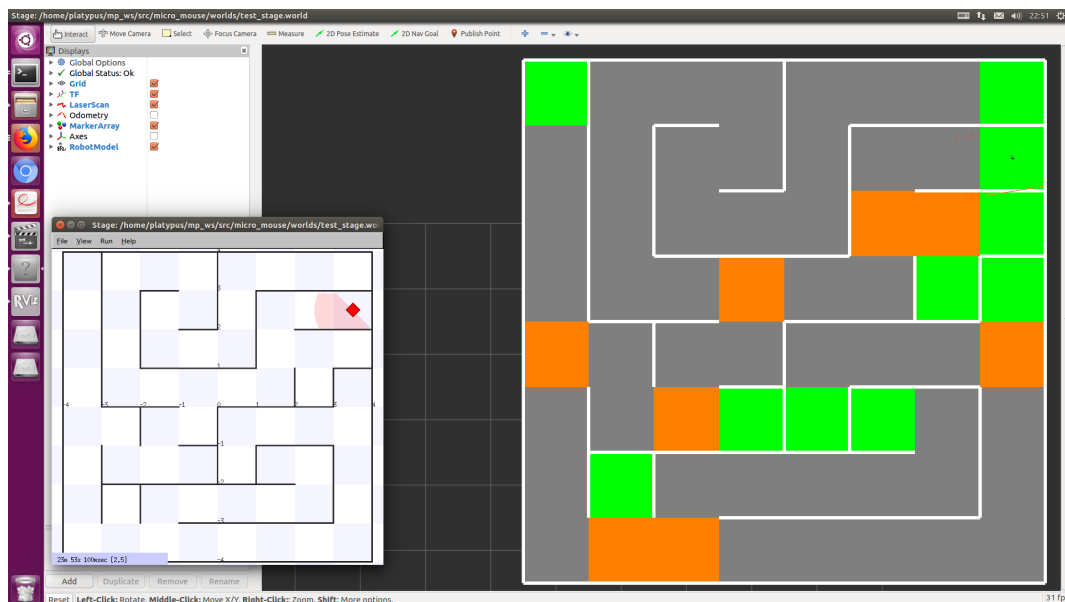
```
$ roscd micro_mouse/script  
$ sudo chmod -R a+x *
```

シミュレータを起動します。以下のコマンドをそれぞれ別端末で実行してください。

ソースコード 3 roscd

```
$ roslaunch micro_mouse startup.launch  
別の端末を開き、  
$ rosrn micro_mouse left_hund.ex.py
```

マウスが動き出し、マッピングが始まれば成功です。しばらく待つと、以下のようになります。



参考文献