



Learn Git and GitHub without any code!

Using the Hello World guide, you'll start a branch, write comments, and open a pull request.

[Read the guide](#)

Branch: **master** ▼

[Find file](#)

[Copy path](#)

[burger_war](#) / [debug.md](#)



gogo5nta Update debug.md

8996df2 now

[1 contributor](#)

[Raw](#)

[Blame](#)

[History](#)



164 lines (150 sloc) 8.27 KB

1. ROSでデバッグ (Visual studio code & ROS & python)

- ・ここではvscode(visual studio code)を使い、ROS上でデバッグする方法について記載する。
- ・更新日 2019年08月19(月) 23:00 by gogo5nta

1. VSCODEの入手

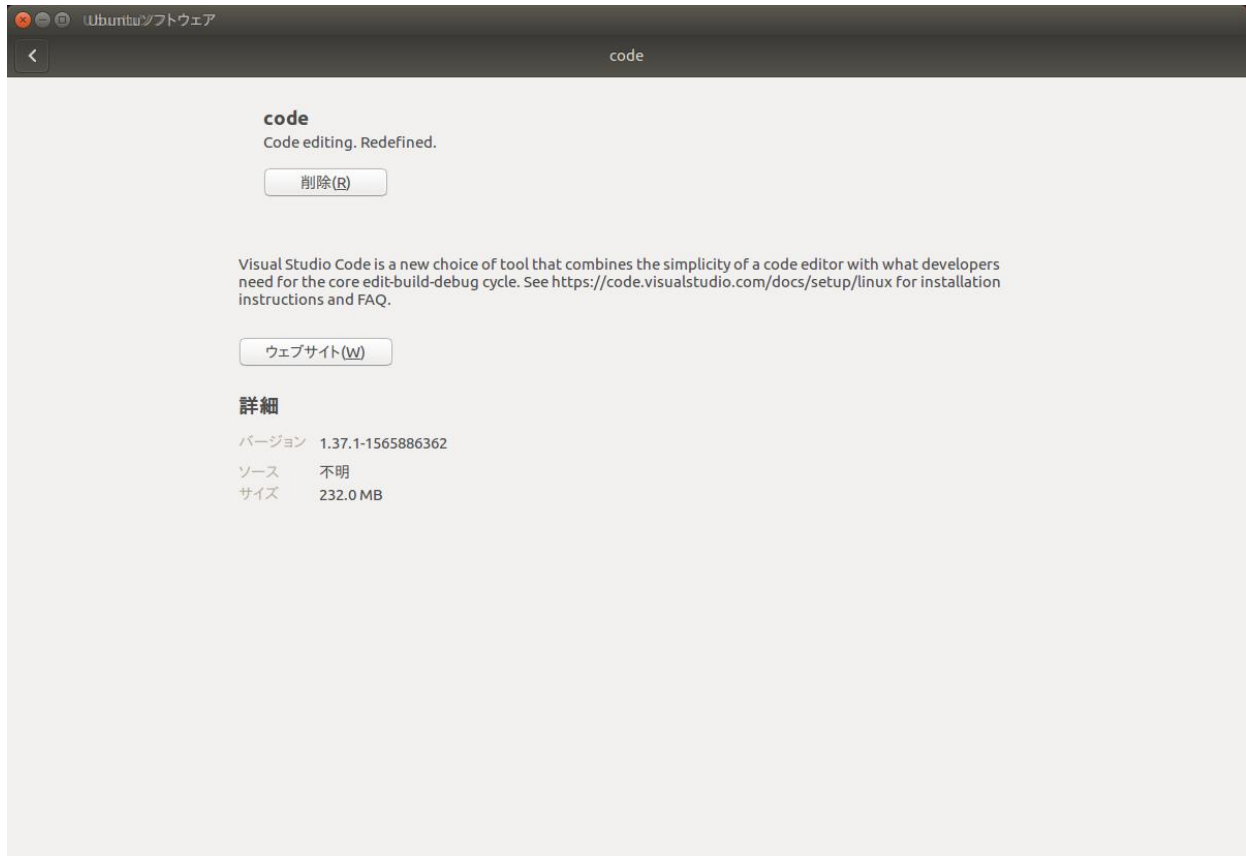
・以下のサイトからvscodeを入手。ubuntuなので.debを選択

Visual Studio Code

The screenshot shows the Visual Studio Code website. The main heading is "Code editing. Redefined." with the subtext "Free. Built on open source. Runs everywhere." Below this are buttons for downloading the .deb package for Debian, Ubuntu, etc., and the .rpm package for Red Hat, Fedora, etc. A link for "Other platforms and Insiders Edition" is also present. The bottom of the website features icons for IntelliSense, Debugging, Built-in Git, and Extensions. An inset image shows the VS Code interface with the Extensions view open, displaying a list of popular extensions like C#, Python, Debugger for Chrome, C/C++, Go, ESLint, PowerShell, and Material-theme.

2. vscodeのインストール

- ・ダウンロードしたファイル(例:/Download/code_xxx.deb)をマウス右ボタンクリック > ソフトウェアのインストールで開くを選択



または、以下のコマンドでもOK

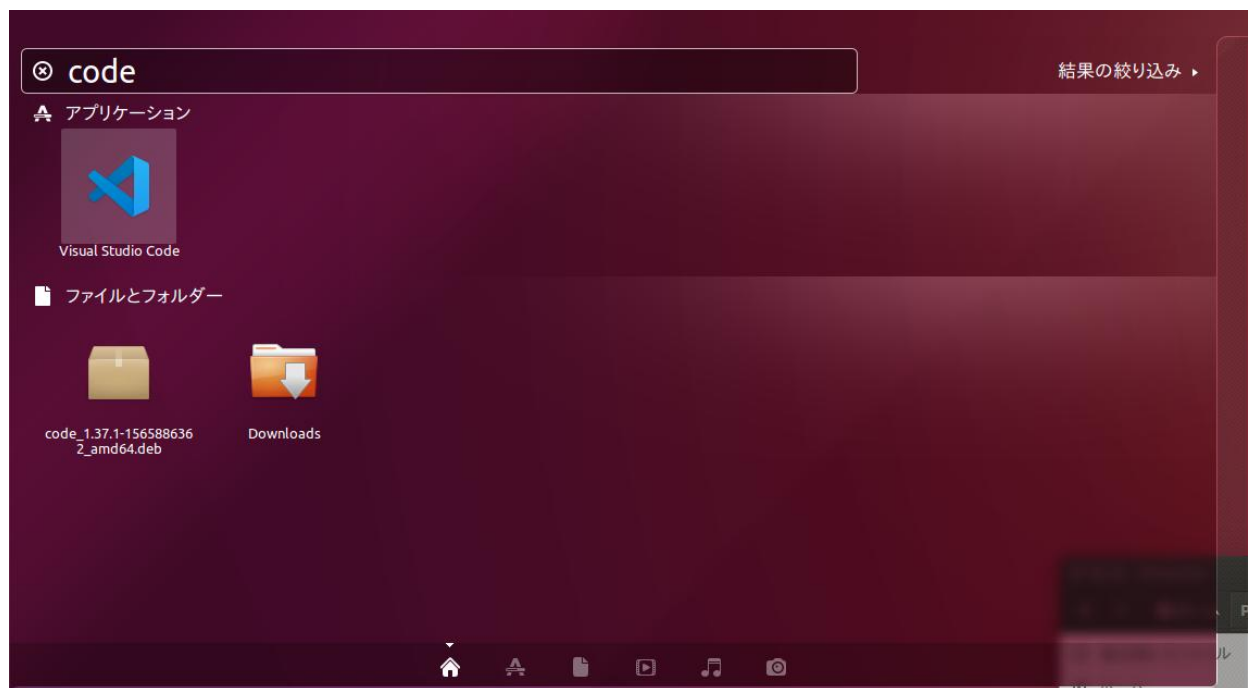
```
$ cd ~/Download
$ sudo dpkg -i code_*.deb
```

3. vscodeの設定

3.1 vscodeの起動

- ・コンピュータの検索でcodeと入力 または、ターミナル上でcodeと入力

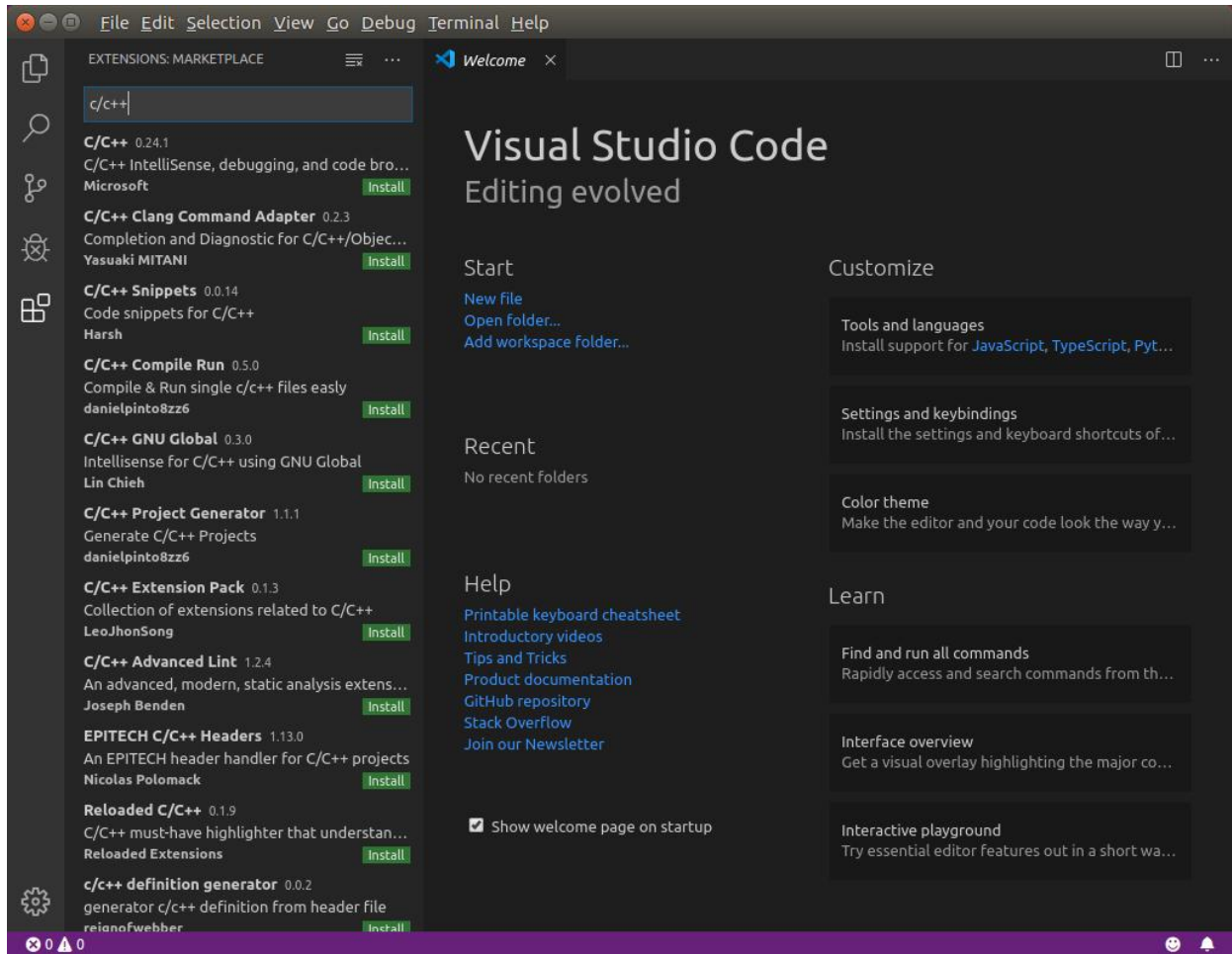
```
$ code
```



3.2 vscodeの設定(初回)

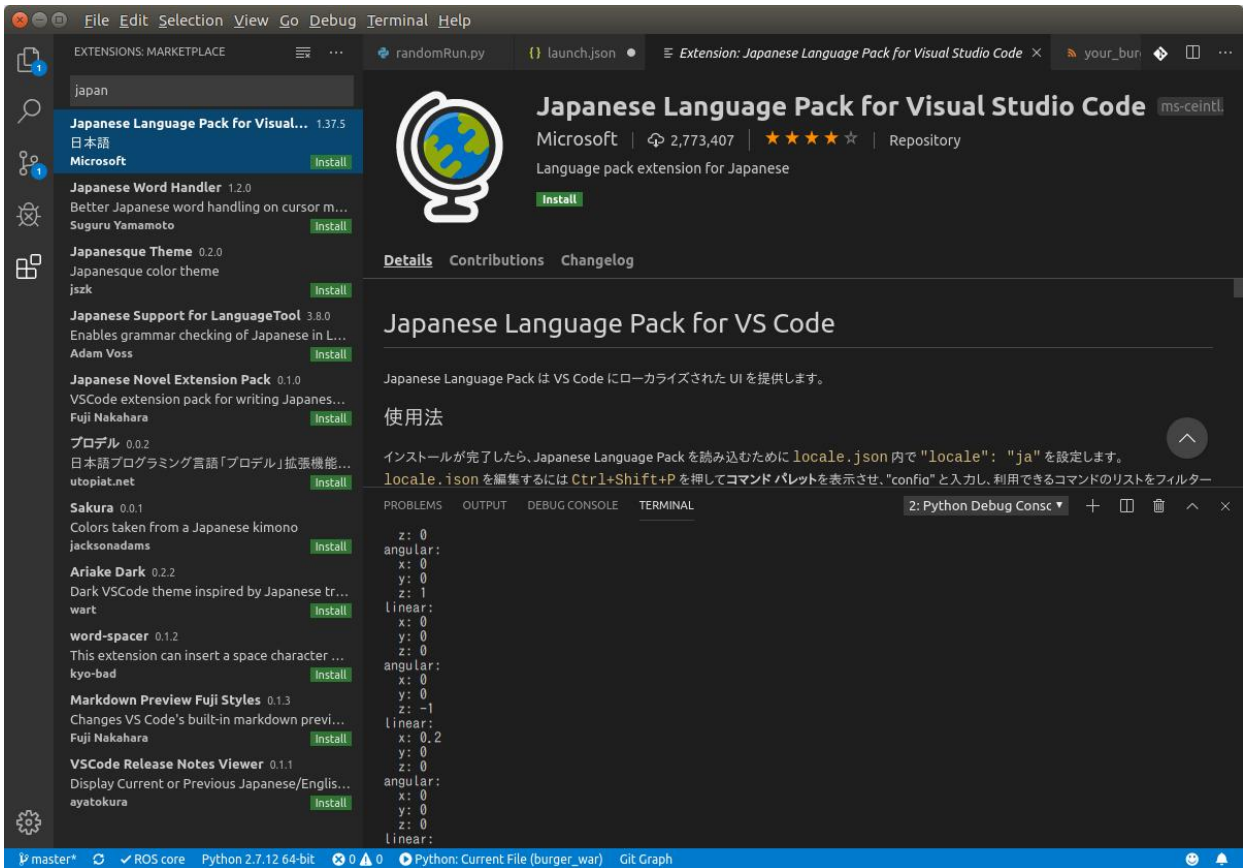
・拡張機能をインストールする。起動したら、左側の5個のアイコンの中で、一番したのアイコンを押し、以下の拡張機能をインストール

- ・C/C++ //C++本体。★必須★
- ・C++ Intellisense //コード補完★必須★
- ・C/C++ Snippets //コード一식을登録する機能。入れなくてもOK
- ・ctags //関数ジャンプ。F1でリスト作成。入れなくてもOK
- ・Python //python本体★必須★
- ・Python for VSCode //python用。入れたほうがいい？
- ・ROS //ROS用★必須★
- ・ROS snippets //ROS用。入れたほうがいい

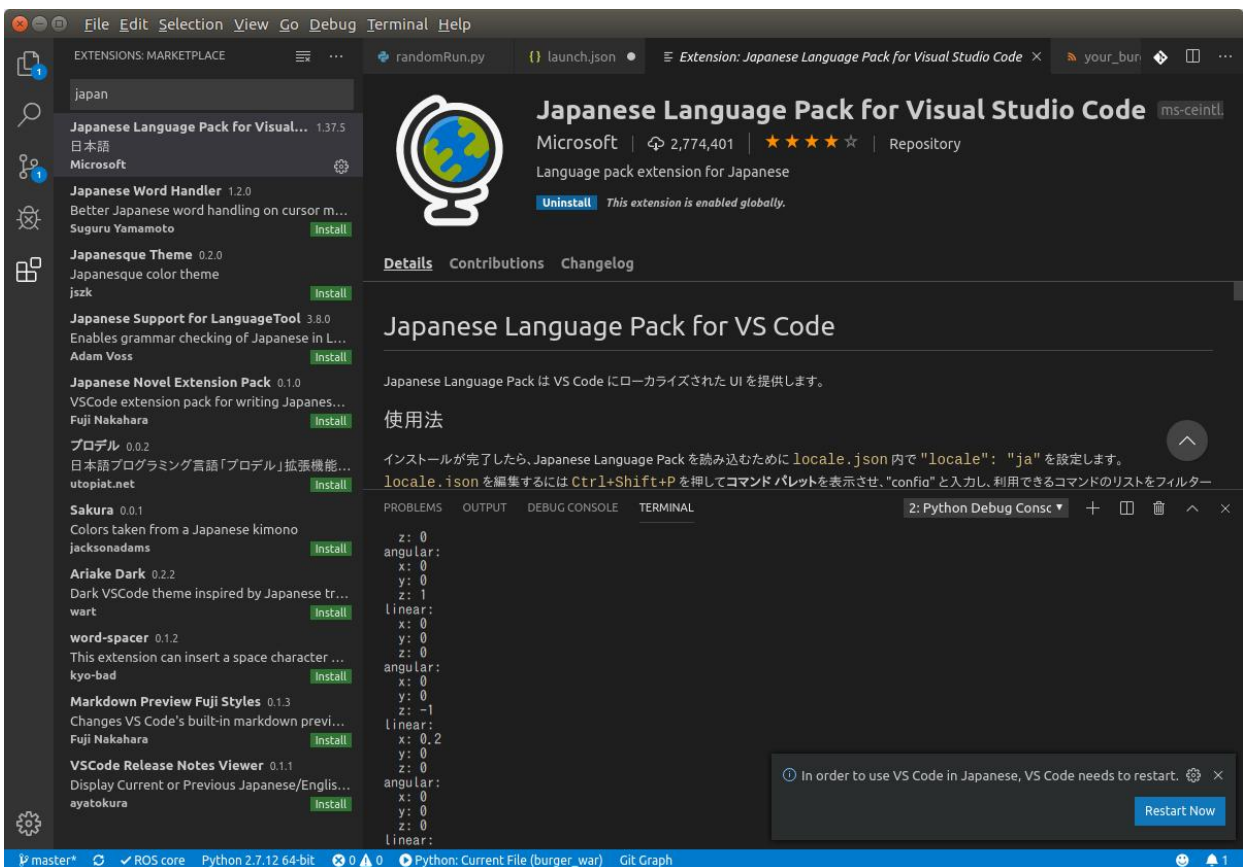


3.3 vscodeの日本語化(必要な方)

・拡張機能で、Japanese Language Pack for VS Codeをインストール



・再起動を聞かれたらYesを選択



・参考 Visual Studio Codeを日本語化する方法

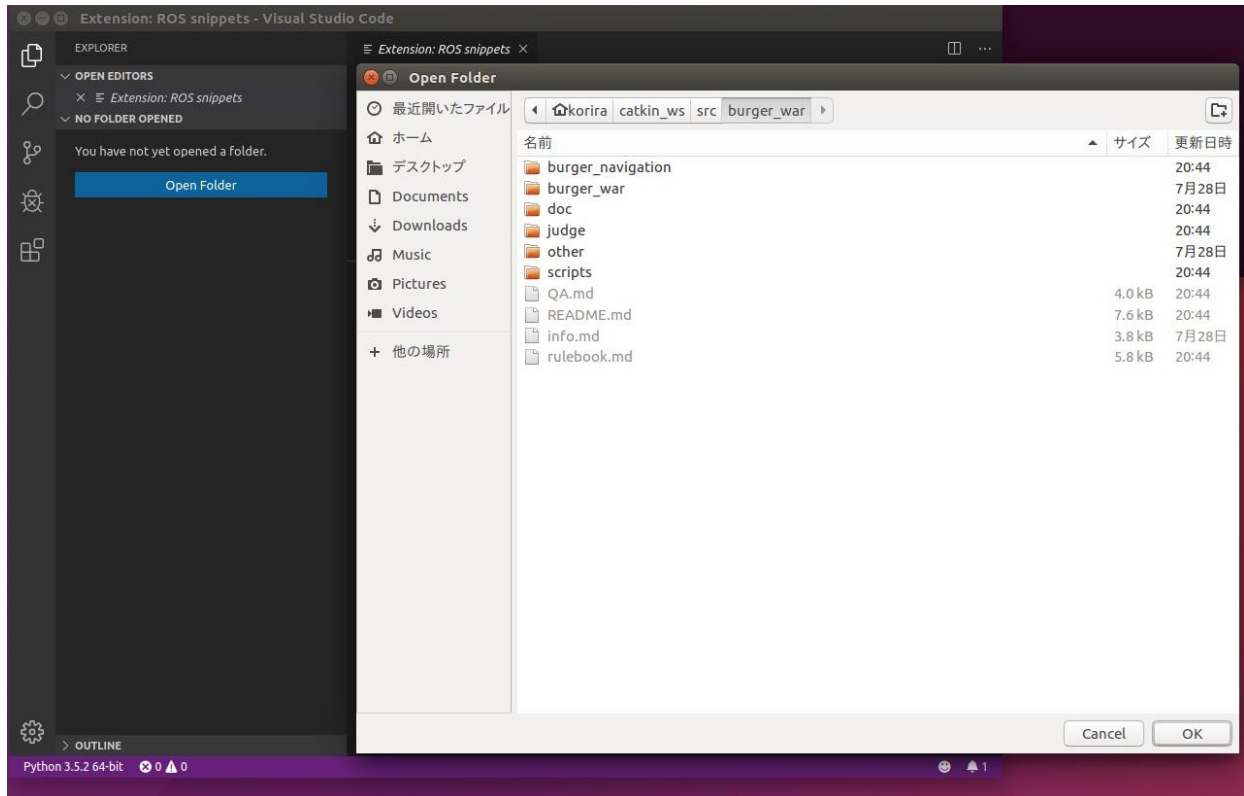
4. ROSで動かす設定

4.1 フォルダの選択

- ・gitで落としてきたburger_warのフォルダを選択。例

```
cd ~/catkin_ws/src
git clone https://github.com/gogo5nta/burger_war
```

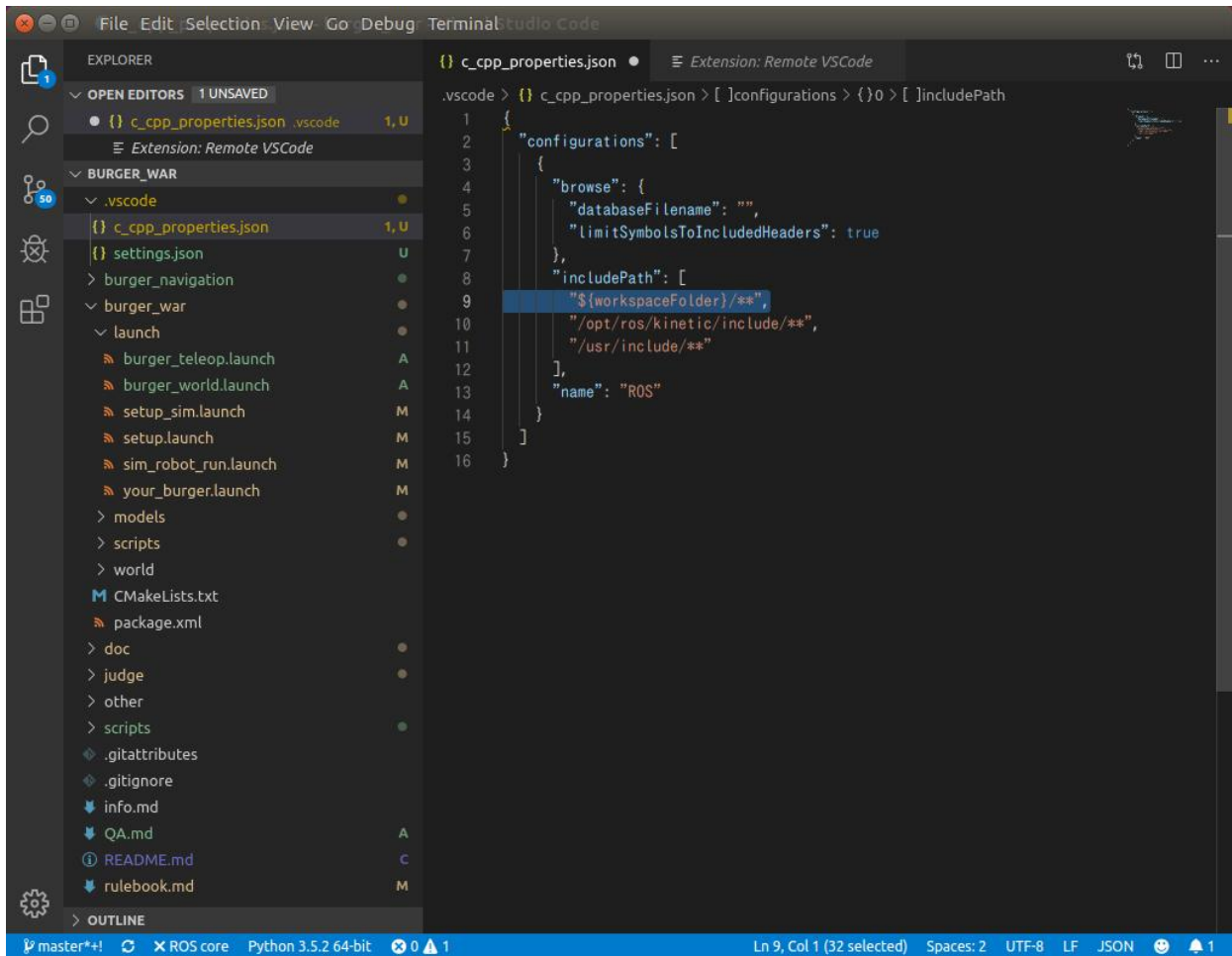
- ・まずは、左側アイコンの一番上を押し、Open Folderを選択
~/catkin_ws/src/burger_warを指定



4.2 c_cpp設定(初回、cpp使用時。pythonのみなら飛ばしてOK)

- ・.vscodeフォルダ内のc_cpp>proprties.jsonを選択し、以下を追加
- ・[VSCodeでROS関連の補完\(C++\)](#)

```
"includePath": [
  "${workspaceFolder}/**",
  "/opt/ros/kinetic/include/**",
  "/usr/include/**",
  .
  .
]
```



4.3 aliasの設定と,terminatorのインストール

- ・複数ターミナルを起動するため、terminatorを事前にインストールする。

```
$ sudo apt-get install terminator
```

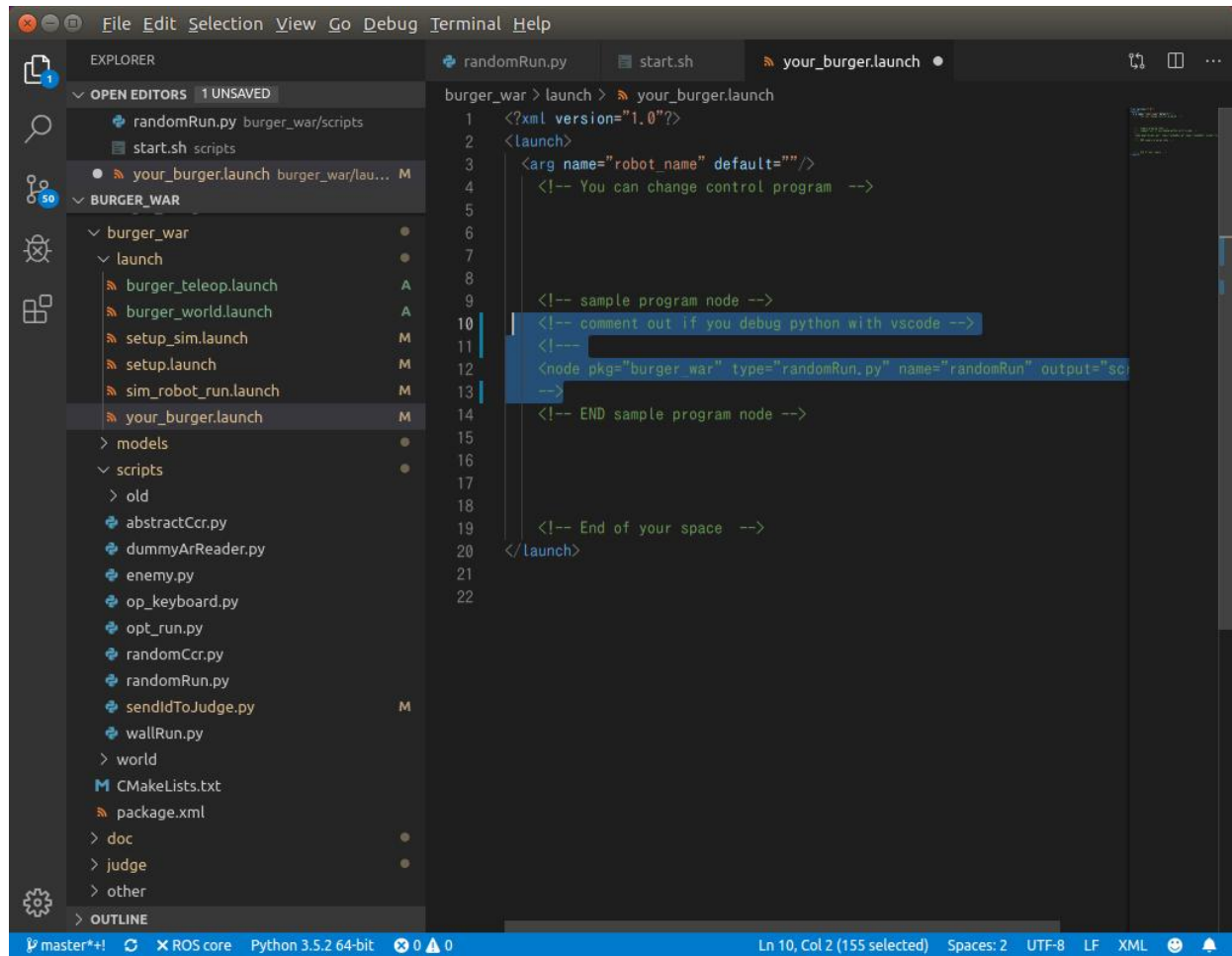
- ・また、短いコマンドでシミュレータ等を起動するため、~/.bashrcファイルをエディタ等で開き、設定を追加後、source ~/.bashrcで読み込ませておく

```
# ROS全般コマンド
alias cw='cd ~/catkin_ws'
alias cs='cd ~/catkin_ws/src'
alias cm='cd ~/catkin_ws && catkin_make'
# robocon2019用コマンド
alias bw='cd ~/catkin_ws/src/burger_war'
alias sim='cd ~/catkin_ws/src/burger_war && bash
scripts/sim_with_judge.sh'
alias start='cd ~/catkin_ws/src/burger_war && bash scripts/start.sh'
# robot位置がリセット。↑のstartで再スタート
alias reset='rosservice call /gazebo/reset_simulation "{}"
```

4.4 python用にlaunchファイルの修正(vscodeでデバッグする場合)

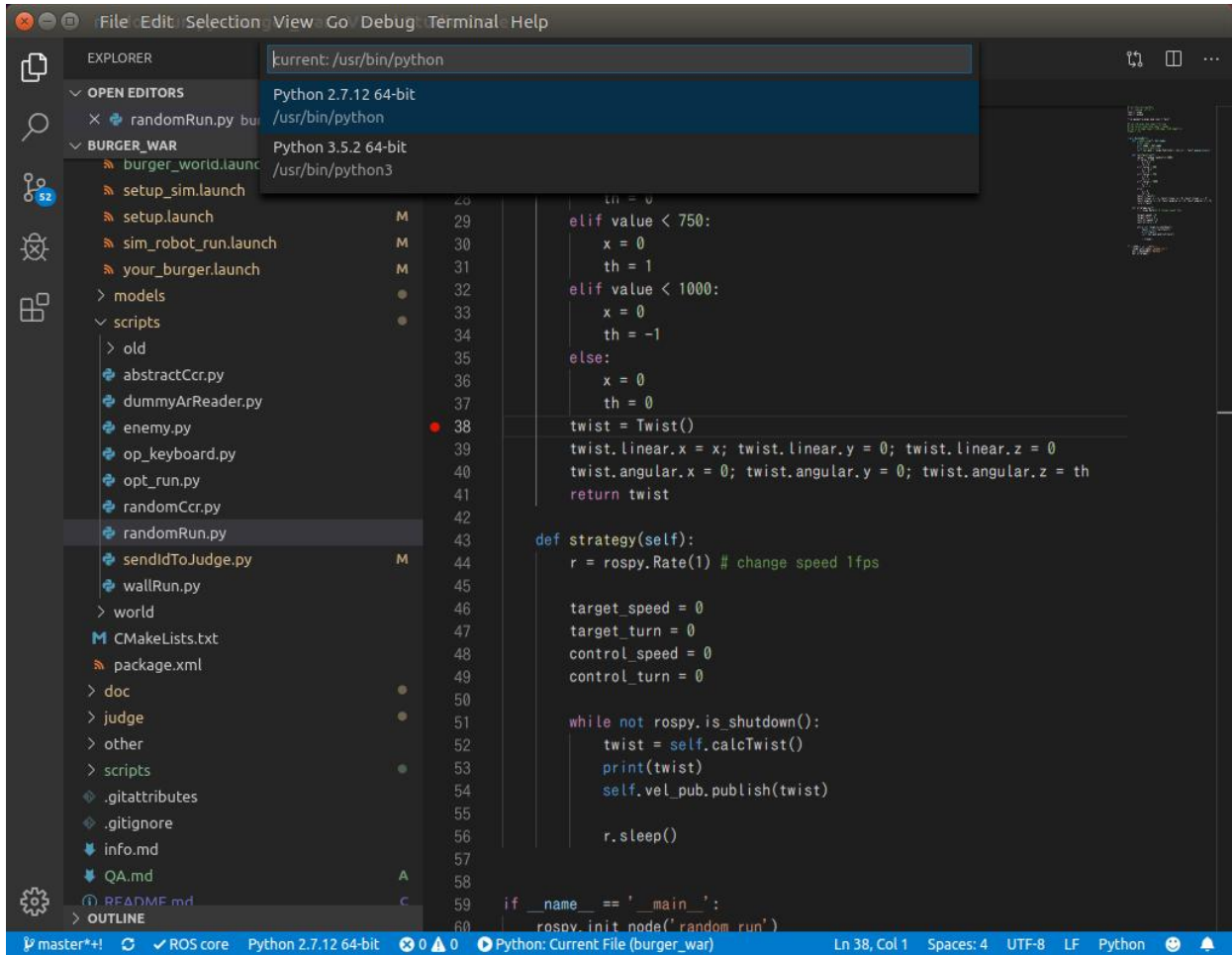
・burger_war/launch/your_burger.launchファイル内のrandomRun.pyをvscodeでデバッグ(起動)するためコメントアウト

```
<!-- sample program node -->
<!-- comment out if you debug python with vscode -->
<!--
<node pkg="burger_war" type="randomRun.py" name="randomRun"
output="screen"/>
-->
<!-- END sample program node -->
```



4.5 python 2.7でデバッグするためのおまじない

- ・Ctrl+Shift+P (同時押し) で"コマンドパレット"というものが開くのでpython select interpreterを検索
- ・python2.7xxxを選択。

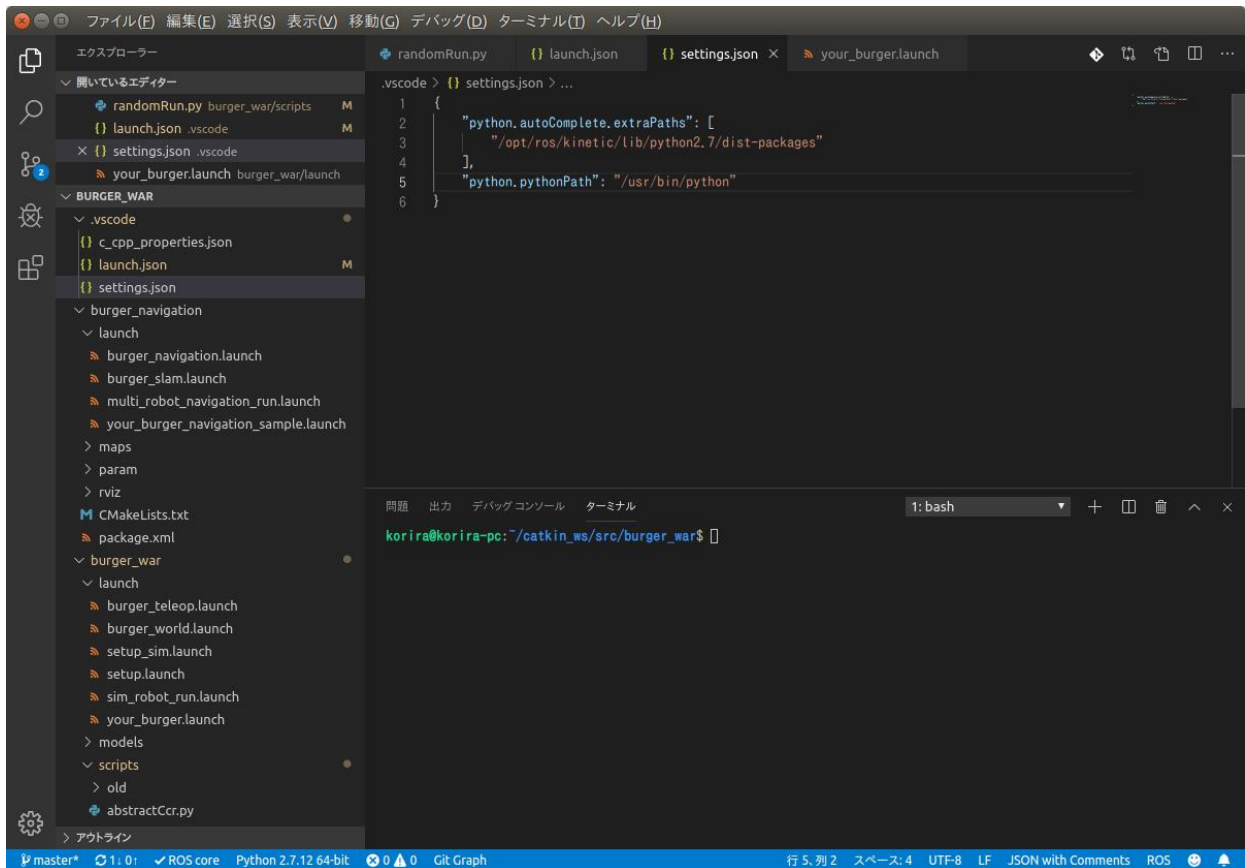


- ・defaultではpython3が選択されているため、vscodeでデバッグすると以下のエラーが発生。解決するまでに半日かかった(T_T)

```
from . message import Message, SerializationError,
DeserializationError, MessageException, struct_I
File "/opt/ros/kinetic/lib/python2.7/dist-packages/genpy/message.py",
line 44, in <module>
    import yaml
ImportError: No module named 'yaml'
```

- ・上記切り替えは、setting.json内の以下を切り替えると同じ

```
"python.pythonPath": "/usr/bin/python"
```

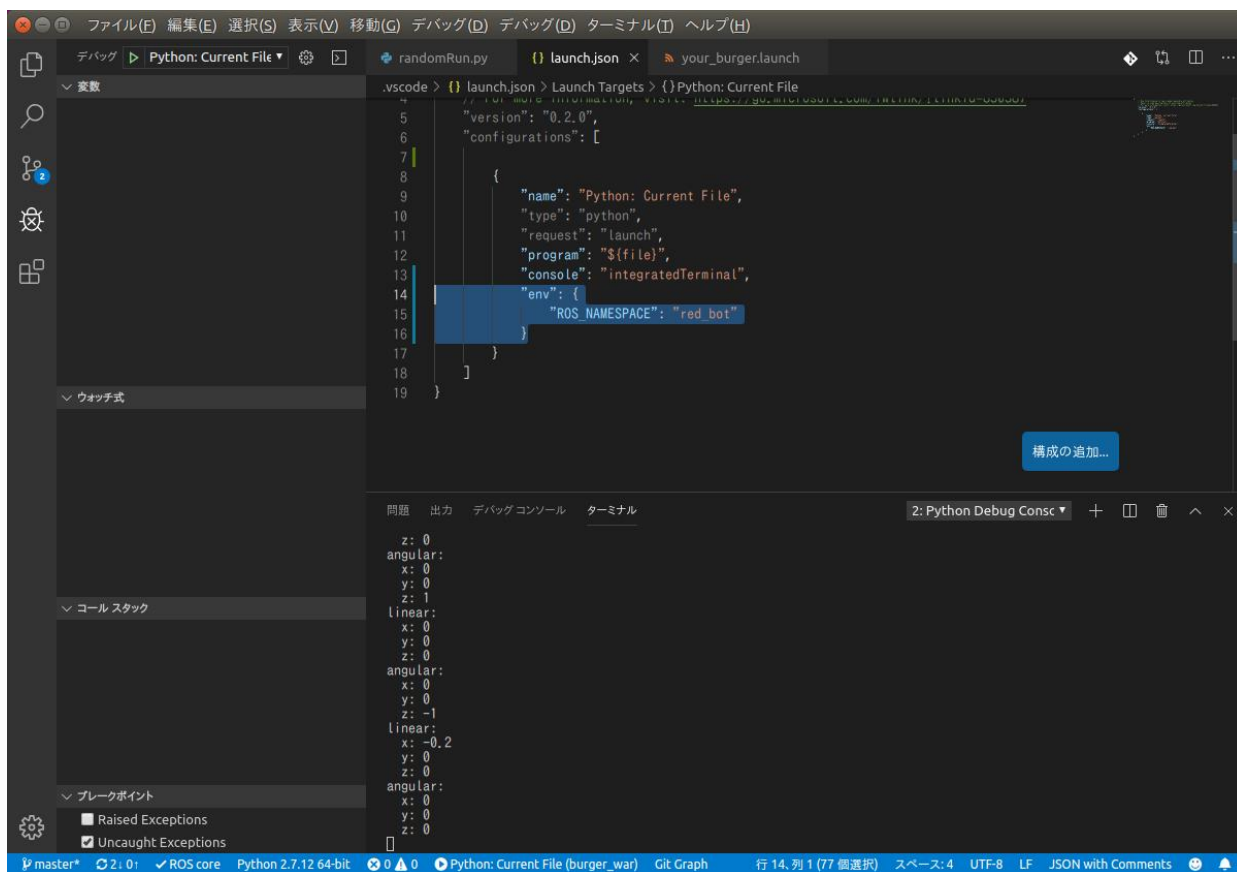


4.6 /red_bot/のNAMESPACEを事前に定義

・vscode上でrandomRun.pyを起動すると、NAMESPACE(red_bot)が与えられないので、 launch.jsonを開き、"env"で設定する。

参考 [VSCode 用の launch.json でデバッグコマンドに環境変数 / 引数 / フラグを渡す設定](#)

```
"configurations": [
  {
    "name": "Python: Current File",
    "type": "python",
    "request": "launch",
    "program": "${file}",
    "console": "integratedTerminal",
    "env": {
      "ROS_NAMESPACE": "red_bot"
    }
  }
]
```

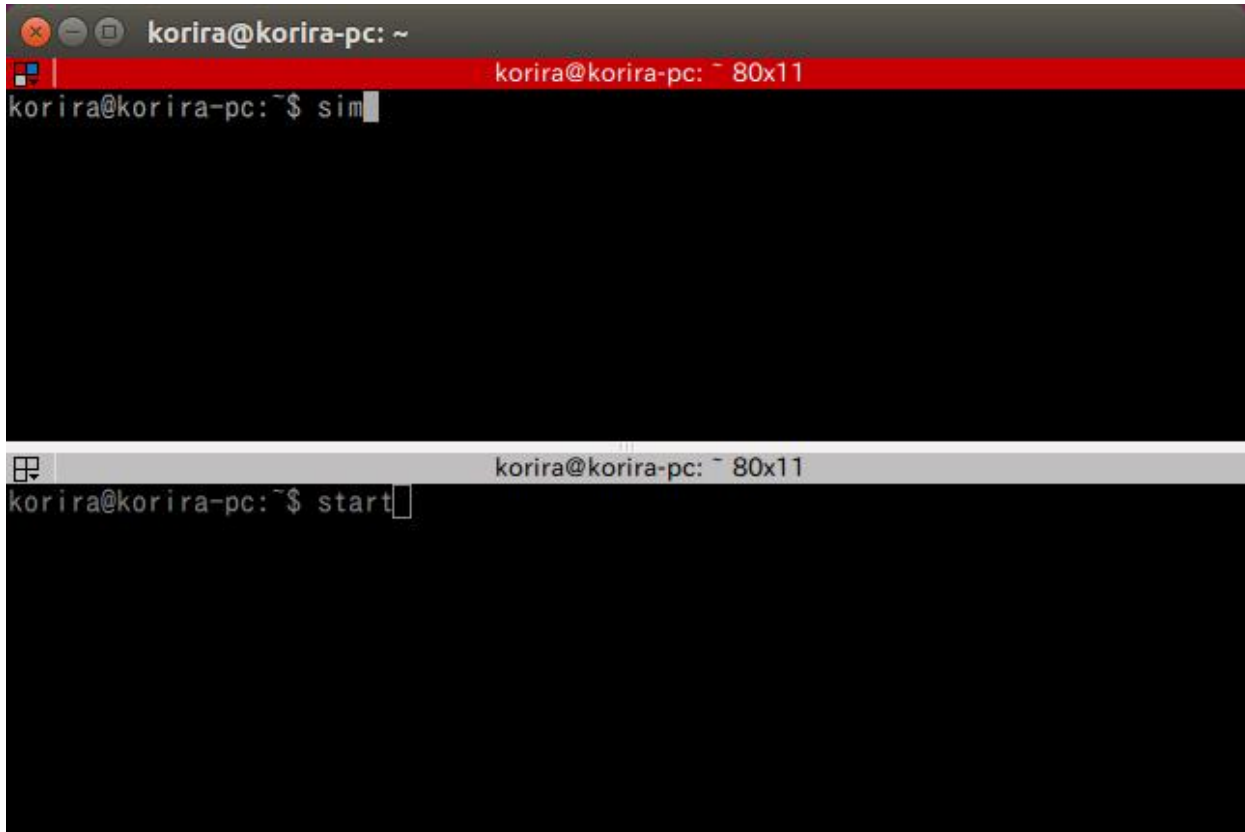


4.7 pythonでデバッグ

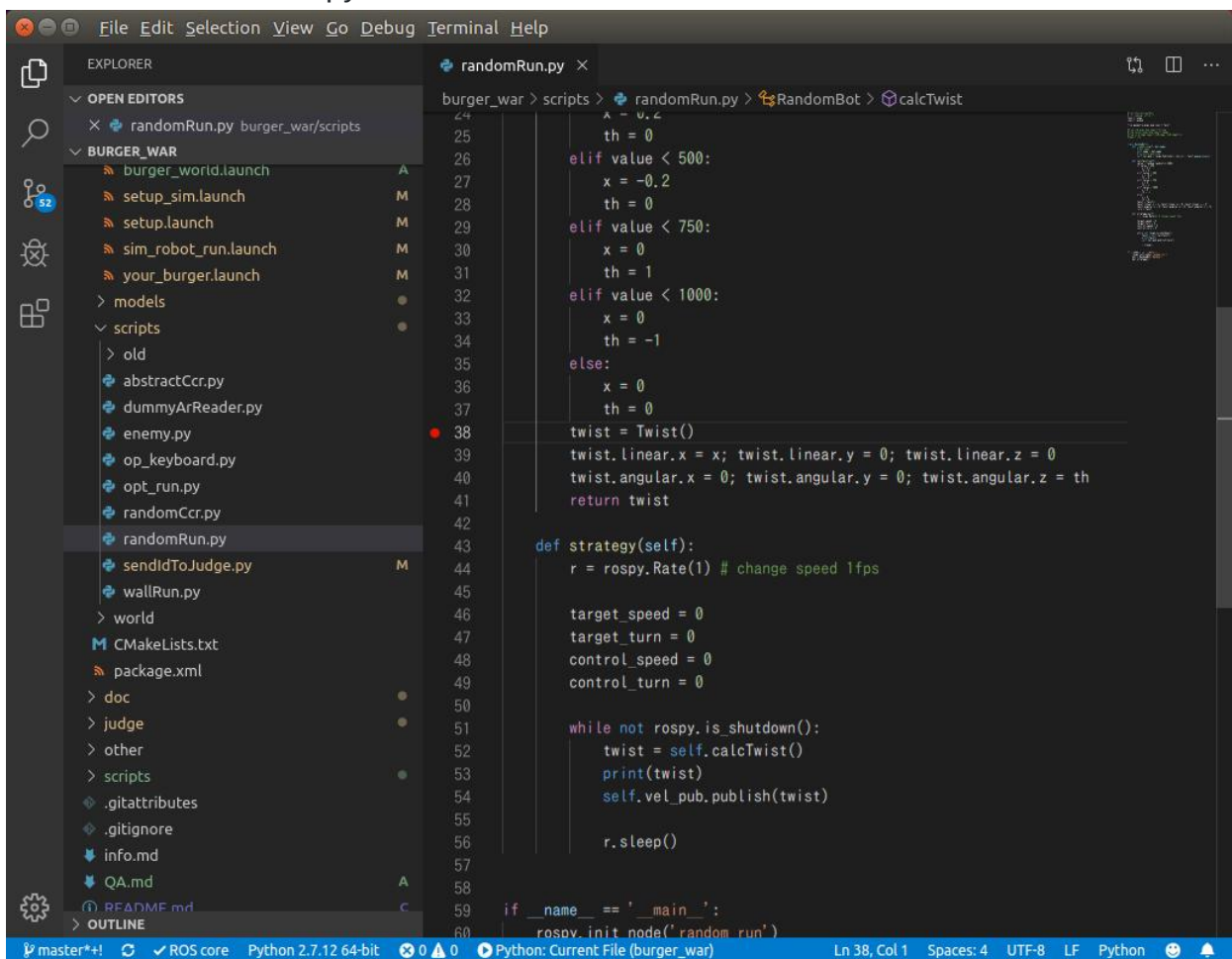
・terminatorを起動し、Ctrl+Shift+O (or Ctrl + Shift + e)でターミナルを2つに分割。

ターミナル1: sim (aliasを使用)

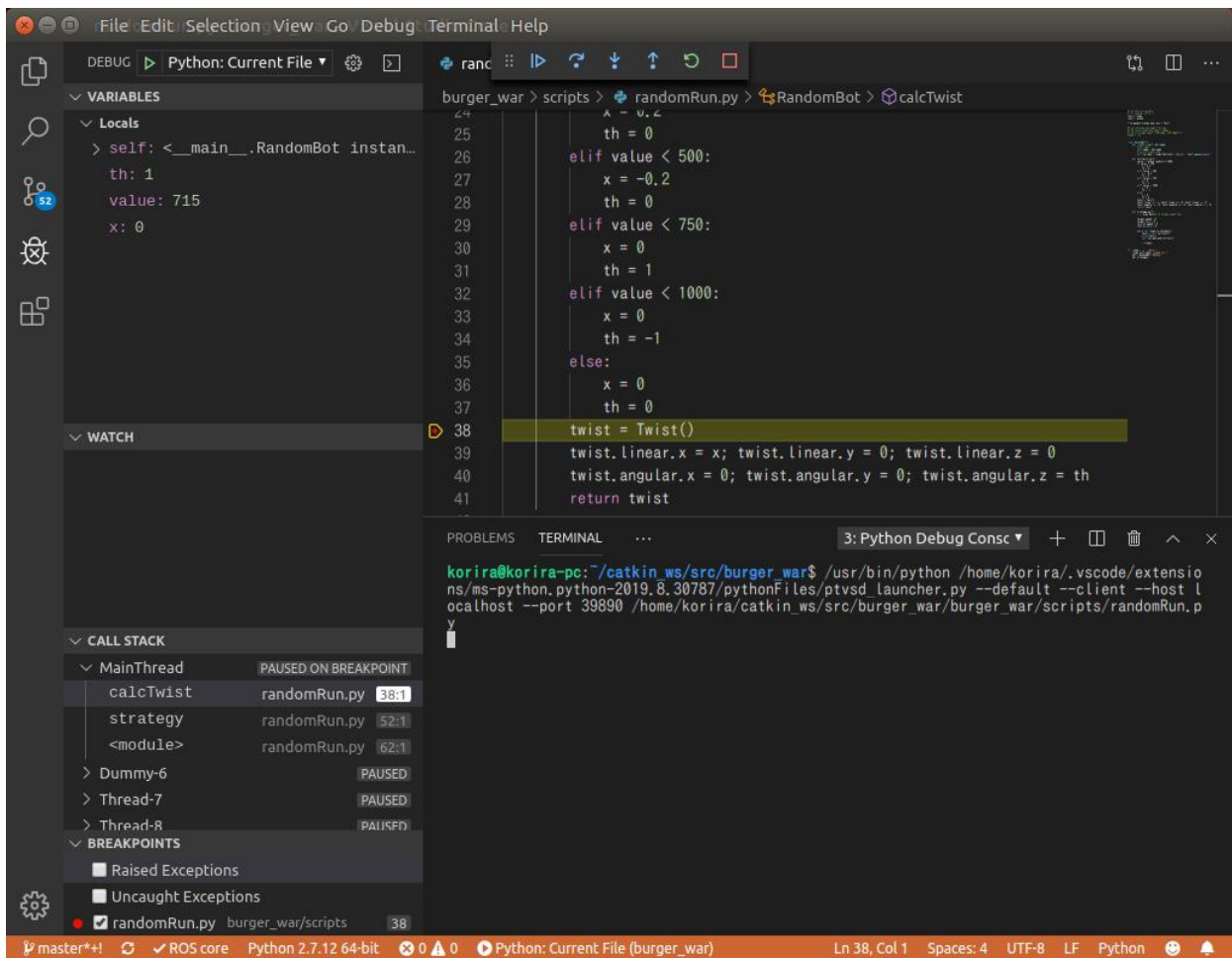
ターミナル2: start (aliasを使用。gazebo起動後に実施)



・vscodeでrandomRun.pyを開き、適当な行にブレークをはる



・左側アイコンの虫眼鏡をおし、DEBUG > python:Current Fileを選択



5. vscodeのgitでソース管理

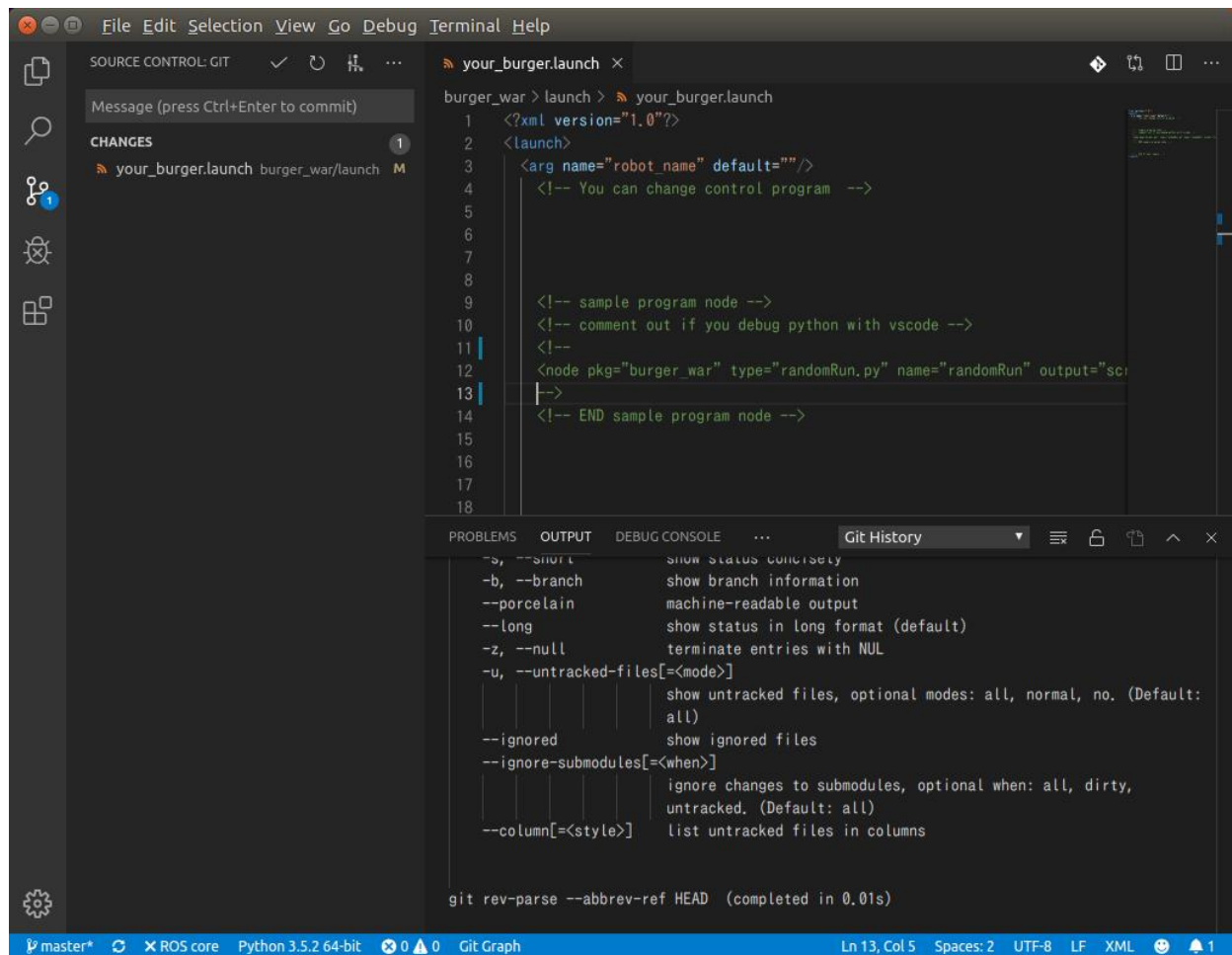
5.1 事前設定

- ・以下のgit全体設定を実施する。

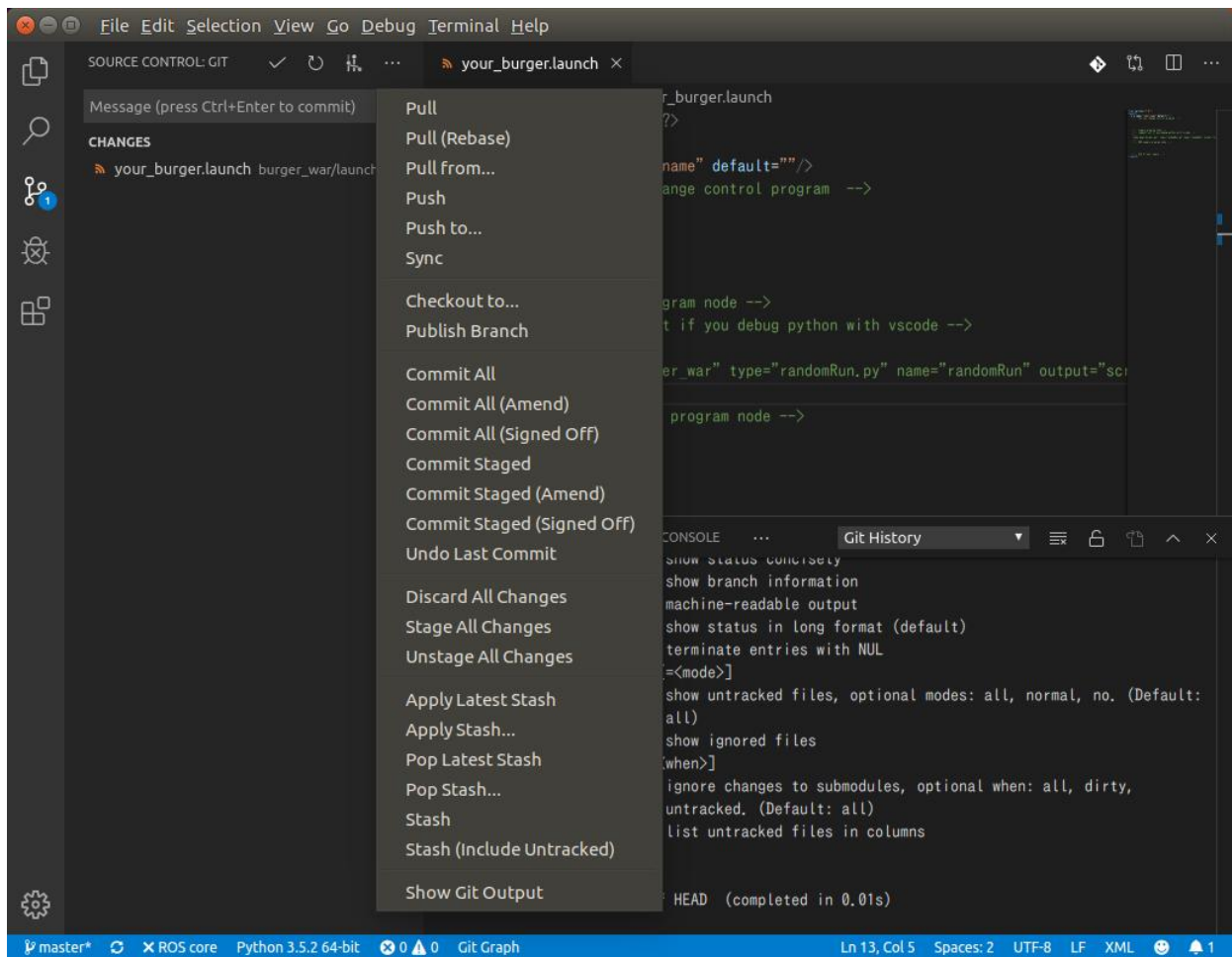
```
$ git config --global user.name 'username'
$ git config --global user.email 'username@example.com'
```

5.2 commit

- ・左アイコン真ん中のツリーアイコンを押し、source管理を実施。コミットはチェックボタンを押すだけ



- ・サーバー上へpushする場合、...のアイコンでpushを選択



・参考サイト

[VSCodeでのGitの基本操作まとめ](#)