

 無料電子ブック

学習

ROS

Free unaffiliated eBook created from
Stack Overflow contributors.

#ROS

.....	1
1: ros	2
.....	2
.....	2
Examples.....	3
.....	3
.....	3
2:	6
.....	6
Examples.....	6
rospy.....	6
3:	7
.....	7
Examples.....	7
rosYaml.....	7
4:	11
.....	11
Examples.....	11
.....	11
.....	12

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [ros](#)

It is an unofficial and free ros ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official ros.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

1: rosをいめる

ロボットオペレーティングシステム**ROS**は、のコンピュータクラスタやプラットフォームでオペレーティングシステムのようなをするロボットソフトウェアのロボットミドルウェアです。

もともとスタンフォードによって2007にスタンフォードAIロボットSTAIRのをけてスタートし、2008から2013までは、ロボットのWillow Garageでされました。2013、ROSのスチュワードシップはOpen Source Robotics Foundationにしました。

ROSは、ソフトウェアがロボットアプリケーションをするためのライブラリとツールをします。ハードウェア、デバイスドライバ、ライブラリ、ビジュアルライザ、メッセージパッシング、パッケージなどをします。ROSはオープンソースのBSDライセンスのでライセンスされています。

、ROSは、からUAV、ヒューマノイドロボット、そしてのROSサポートされたセンサー [なセンサーのリスト](#) をして、100のロボット [フルロボットリスト](#) をしています。サービスロボットアプリケーションのコミュニティでもありますが、そのはロボットをむのアプリケーションにもできます。した、パス、モーショントラッキングなどのアプリケーションは、はにであったか、またはコストがすぎるロボットアプリケーションをすることができます。

、ROSはUnixベースのプラットフォームでのみします。ROSのソフトウェアはにUbuntuおよびMac OS Xシステムでテストされていますが、ROSコミュニティはFedora、Gentoo、Arch Linux およびそののLinuxプラットフォームのサポートにしています。のところ、ROSコーディングは [クライアントライブラリ](#) をっていればどのプログラミングですることでもできますが、はなC++とPythonのサポートをすることにをてています。

ROSは、10のリリースROS *Kinetic*をします。

ROSおよびROSコミュニティのりみについては、 [http //www.ros.org/](http://www.ros.org/)を [ご覧ください](#)。

バージョン

ロスディストロ	サポートされているUbuntuのバージョン	
キネティックカメ	15.10,16.04	2016523
タートル	14.04、14.10、15.04	2015-05-23
インディゴイグルー	13.10、14.04	2014-07-22
ハイドロメデューサ	12.04、12.10、13.04	2013-09-04
グルーヴィーガラパゴス	11.10,12.04,12.10	2012-12-31
フエルテタートル	10.04,11.10,12.04	2012-04-23

ロスディストロ	サポートされているUbuntuのバージョン	
エミ	10.04、 10.10、 11.04、 11.10	2011-08-30
ダイヤモンドバック	10.04、 10.10、 11.04	20110302
Cタートル	9.04、 9.10、 10.04、 10.10	2010-08-02
ボックスタートル	8.04	20100302

Examples

インストール

ターゲットマシンにじて、サポートされている**ROS**バージョンをするがありますまたはその。
 ROSのインストールはROSのwikiによくかれています、つけにくいかもしれません。 ROSのバージョン、ターゲットプラットフォームとアーキテクチャ、およびインストールガイドのリンクはのとおりです。

ROSバージョン	プラットフォーム	アーチ		インストールガイドリンク
キネティック	Ubuntu 16.04Xenial	amd64 / i386 / armhf	サポートされる	キネティック - ゼニア - ガイド
	Ubuntu 15.10ウィリー	amd64 / i386	サポートされる	キネティック - ウィリー - ガイド
	Debian 8Jessie	amd64 / arm64	サポートされる	キネティック - ジェシー - ガイド
	OS X	-		キネティック - ガイド
	Gentoo	-		Kinetic-Gentoo-guide
	OpenEmbedded / Yocto	-		キネティックヨコガイド

の...

ハローワールドパブリッシャー

ワークスペースをする

```
mkdir -p ~/catkin_ws/src
cd ~/catkin_ws/src
```

```
catkin_init_workspace
```

ワークスペースをする

```
cd ~/catkin_ws/  
catkin_make
```

セットアップファイルをする

```
source devel/setup.bash
```

なをつhello_worldというのしいパッケージをする

```
catkin_create_pkg hello_world std_msgs rospy roscpp
```

srcディレクトリにし、**talker.cpp**というしいファイルをしします。

```
cd hello_world/src  
touch talker.cpp
```

しいファイルをし、このコードをりけて「こんにちは」のメッセージをしてください

```
#include "ros/ros.h"  
#include "std_msgs/String.h"  
  
#include <sstream>  
  
int main(int argc, char **argv)  
{  
    ros::init(argc, argv, "talker");  
  
    ros::NodeHandle n;  
  
    ros::Publisher chatter_pub = n.advertise<std_msgs::String>("chatter", 1000);  
  
    ros::Rate loop_rate(10);  
  
    int count = 0;  
    while (ros::ok())  
    {  
        std_msgs::String msg;  
  
        std::stringstream ss;  
        ss << "hello world " << count;  
        msg.data = ss.str();  
  
        ROS_INFO("%s", msg.data.c_str());  
  
        chatter_pub.publish(msg);  
  
        ros::spinOnce();  
  
        loop_rate.sleep();  
        ++count;  
    }  
}
```

```
}  
  
return 0;  
}
```

パッケージディレクトリのルートに戻る

```
cd ..
```

これらのをCMakeLists.txtに/コメントをします。

```
catkin_package(  
  INCLUDE_DIRS include  
  LIBRARIES hello_world  
#  CATKIN_DEPENDS roscpp rospy std_msgs  
#  DEPENDS system_lib  
)  
  
include_directories(include ${catkin_INCLUDE_DIRS})  
  
add_executable(talker src/talker.cpp)  
target_link_libraries(talker ${catkin_LIBRARIES})  
add_dependencies(talker hello_world_generate_messages_cpp)
```

のルートに戻る

```
cd ..
```

しいパブリッシャーをする

```
catkin_make
```

しいパッケージとパブリッシャーをつようにセットアップファイルをソースにします

```
source devel/setup.bash
```

ROSをする

```
roscore
```

roscoreをしたままにして、しいターミナルタブ/ウィンドウでパブリッシャーをする

```
roslaunch hello_world talker
```

パブリッシャーをしたままにして、のしいタブ/ウィンドウでをエコーします

```
rostopic echo /chatter
```

オンラインでrosをいめるをむ <https://riptutorial.com/ja/ros/topic/7287/rosをいめる>

2: パッケージの

き

このチュートリアルでは、ROSでパッケージを作るをします。パッケージは、srcディレクトリにあります。パッケージディレクトリにはCMakeLists.txtとpackage.xmlファイルがです。

Examples

rospyを使ってパッケージを作る

workspace_nameというのworkspace_nameがにホームディレクトリにされているとすると、のコマンドラインをすることによってpackage_nameというのパッケージをできます。

```
$ cd ~/workspace_name/src/  
$ catkin_create_pkg package_name rospy
```

オンラインでパッケージのをむ <https://riptutorial.com/ja/ros/topic/8314/パッケージの>

3: ローズランチ

'ノードはパッケージからノード'でなければならぬ'ノードはパッケージからフォーム'
は ""と ""とっていますが、されたシャットダウンをうはしていません。

Examples

ros ノードをし、Yamlファイルからパラメータをロードする

roslaunchは、ROSノードのとをするなツールです。1つの "*" .launch"ファイルをととしてとりま
す。

このでは、のものをしますこのでされるように。したがって、これらのコマンドをどのようにし
てにすることができますか

```
roscd stereo_camera
rosparam load marvin_cameras.yaml
roslaunch stereo_camera stereo_camera __name:=bumblebeeLeft
roslaunch stereo_camera stereo_camera __name:=bumblebeeCenter

roslaunch openni_launch_marvin kinect_left.launch
roslaunch openni_launch_marvin kinect_center.launch
```

まず、 break up these commands in pieces みましよう。 *roscd*、 *rosparam*、 *roslaunch*
の4つのコマンドがです。さあめましよう

これはroslaunchがなのです。、これらのコマンドはすべてone and only roslaunch file in ROSま
れていone and only roslaunch file in ROS。これらのコマンドをしてにびすために、この "
solution.launch"ファイルをするだけです。

するには、ファイルはXMLフォーマットにづいています。ROSのなファイルは "
basic_example.launch"とけられ、"roslaunch_example"というのROSパッケージにまれています
。

```
<launch>
</launch>
```

このファイルをするコマンドはのとおりです。

```
$ roslaunch roslaunch_example basic_example.launch
```

にう

```
$ roslaunch package_name launch_file_name.launch
```

ファイルにはコマンドがまかれていないので、もされません。

ファイルにノードをめる

インストールされているROSパッケージのROSノードは、ファイルでびしです。そのためには、ノードをむパッケージとパッケージにされているをするがあります。えは

```
roslaunch stereo_camera stereo_camera __name:=bumblebeeLeft
roslaunch stereo_camera stereo_camera __name:=bumblebeeCenter
```

stereo_camera、ノードがパッケージで stereo_camera、されたは、それがだあり
__name:=bumblebeeLeft と __name:=bumblebeeCenter。

これらのノードをするには、のをするがあります。

```
<launch>
  <node name="$(arg name)" pkg="stereo_camera" type="stereo_camera" output="screen">
    <param name="name" value="bumblebeeLeft" />
  </node>

  <node name="$(arg name)" pkg="stereo_camera" type="stereo_camera" output="screen">
    <param name="name" value="bumblebeeCenter" />
  </node>
</launch>
```

このファイルをすると、2つのノードがされます。

ROSノードにパラメータをする

わかるように、ノードのパラメータ「」をのようになしました。

```
<param name="name" value="bumblebeeCenter" />
```

には、でしたようになだけくのパラメータをし、のをするのではなく、 "\$(arg parameter_name)" びすことでできます。

の

on the terminal (~/.ros) をるがあるはoutputタグを "screen"にし、ログファイルを (~/.ros)にするには "log"をします。

ROS ファイルにのROS ファイルをめる

ここにされているように、タグをすると、のroslaunch XMLファイルをのファイルにインポートできます。これは、タグやをむドキュメントののスコープにインポートされます。インクルードファイルのすべてのコンテンツは、タグをいてインポートされます。タグはファイルにのみいます。

したがって、これらのコマンドをすべてするには

```
roslaunch openni_launch_marvin kinect_left.launch
roslaunch openni_launch_marvin kinect_center.launch
```

ファイルから、のをするだけです。

```
<include file="$(find openni_launch_marvin)/launch/kinect_left.launch" />
<include file="$(find openni_launch_marvin)/launch/kinect_center.launch" />
```

ROSファイルのパッケージにroscdする

たちがインクルードしたいファイルをつけるために、なパスをするはありません。わりに、roslaunchは"`$(find package_name)`"ディレクティブをします。ここでは`relative to the package racine`ファイルができます。

のでは、"`kinect_center.launch`"というファイルが"`openni_launch_marvin / launch /`"フォルダにあるとしました。

YAMLファイルからパラメータをロードする

ROSのYAMLファイルからパラメータをロードするために、ROSは"`rosparam`"タグをします。wikiにされているように、「このタグは、ROSパラメータサーバからパラメータをロードしてダンプするための`rosparam` YAMLファイルのをにします。パラメータはプライベートネームのようにわれます。

このタグをして、のをして、YAMLファイルをファイルにロードできます。

```
<rosparam command="load" file="$(find marvin_cameras)/config/marvin_cameras.yaml" />
```

のように、はYAMLファイル"`marvin_cameras.yaml`"が"`marvin_cameras / config /`"フォルダにあるとしました。

すべてのをみてる

ファイルのを々にしたので、きなファイル"`solution.launch`"でそれらを見てみましょう。

ソリューション。

```
<launch>

  <rosparam command="load" file="$(find marvin_cameras)/config/marvin_cameras.yaml" />

  <node name="$(arg name)" pkg="stereo_camera" type="stereo_camera" output="screen">
    <param name="name" value="bumblebeeLeft" />
  </node>

  <node name="$(arg name)" pkg="stereo_camera" type="stereo_camera" output="screen">
    <param name="name" value="bumblebeeCenter" />
  </node>

  <include file="$(find openni_launch_marvin)/launch/kinect_left.launch" />
  <include file="$(find openni_launch_marvin)/launch/kinect_center.launch" />
```

```
</launch>
```

ここでは、すべてのコマンドをしてにするためののroslaunchファイルがあります。

オンラインでローズランチをむ <https://riptutorial.com/ja/ros/topic/7361/ローズランチ>

4: ワークスペースの

き

このチュートリアルでは、ワークスペースを作るをします。ワークスペースとは、ROSコードのセットがするのディレクトリです。1つはのROSワークスペースをつことができますが、に1つだけすることはです。

Examples

ワークスペースの

ワークスペースを作るには、でのコマンドをするがあります。

```
$ mkdir -p ~/workspace_name/src
$ cd ~/workspace_name/src
$ catkin_init_workspace
$ cd ~/workspace_name/
$ catkin_make
```

のコマンドは、workspace_nameというのworkspace_nameしworkspace_name。ワークスペースをしたら、そのワークスペースをしてすることがです。

```
$ source ~/workspace_name/devel/setup.bash
```

オンラインでワークスペースのをむ <https://riptutorial.com/ja/ros/topic/8313/ワークスペースの>

クレジット

S. No		Contributors
1	rosをいめる	ckirksey3 , Community , Photon , Vtik
2	パッケージの	Imiguelvargasf , Michael
3	ローズランチ	ensonic , Vtik
4	ワークスペースの	Imiguelvargasf