

ROS introduction

Powering the world's robots



機器人開發

Human-Computer Interaction

人機互動



Middleware

中介軟體層



Firmware & Driver

韌體與驅動程式



Hardware

硬體



Visualization

視覺化



系統整合



Environment Perception

環境感知



Motion Planning

運動規劃



High Level Applications

使用者應用程式



Communication

通訊



Sensor Fusion

感測器整合



ROS系統介紹

希望在開發機器人上 可以擁有爆發性的成長

機器人操作系統(Robot Operating System)，
是專為機器人軟體開發所設計出來的一套電腦作業系統架構。
它是一個開源的元級作業系統，提供類似於作業系統的服務，
包括硬體抽象描述、底層驅動程序管理、共用功能的執行、程序間消息傳遞、程序發行包管理，它也提供一些工具和庫用於獲取、建立、編寫和執行多機融合的程序。

Robot Operating System

- 開源的作業系統，採用 BSD 授權條款
- 屬於「元操作系統 (meta-operating system)」
- 創造者為 Willow Garage 組織
- 目前由 open source robotics foundation 維護與管理
- 最大目標是希望增加「代碼重複性」的使用



Plumbing 探测



Tools 工具



Capabilities 功能



Ecosystem 生態系統

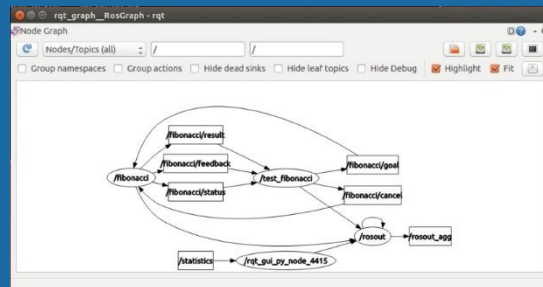


Robot Operating System

探測、工具、功能、生態系

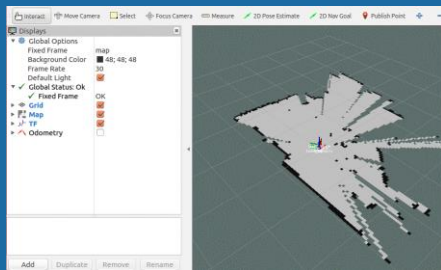
探測

ROS可以查看動態圖(Dynamic graph)
確認是由哪一個節點發布訊息



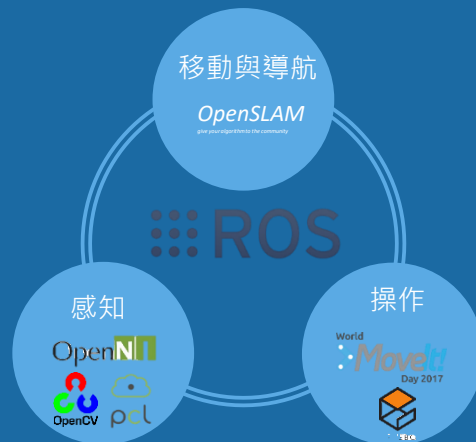
工具

ROS 擁有 Rqt 、Rviz 等工具讓開發者能夠快速地收集開發相關系統所需要的數據



功能

ROS 共有三大功能區塊
移動與導航、感知、操作

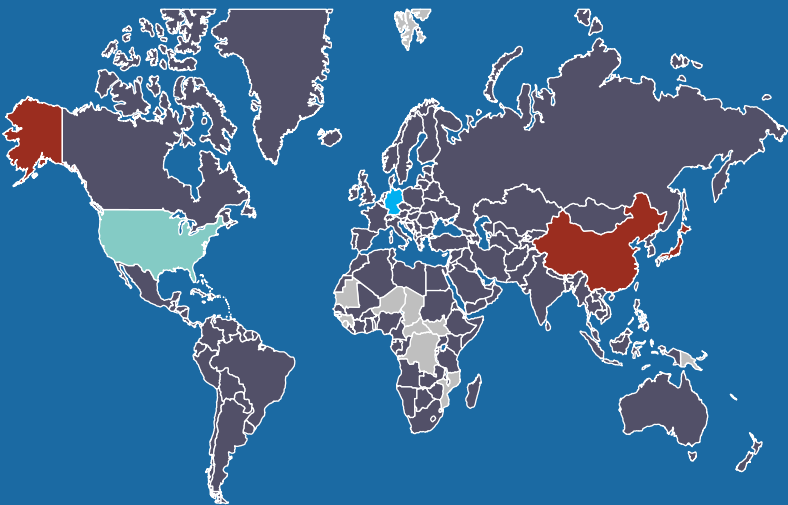




生態系統

2016 年度社群指標報告

- 年增長率：21%
- 維基百科資料數：17058
- 維基百科每日編輯數：14.7/天
- 維基百科瀏覽人數：44,979/天



1 85.023



BOSCH

dji

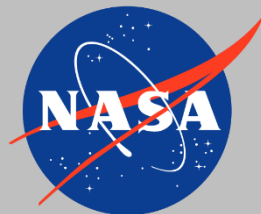
使用 ROS 系統的 Rviz
與 AprilTag，以便追蹤
無人機位置



TOYOTA

Intel

將深度相機放置 NVIDIA
Jetson TX 控制板上，並
使用 ROS 系統操作深度
相機



BOSCH

2014 年與 Google 的
Tango 合作，將
PointCloud 數據 導出



TOYOTA

Toyota Research
Institute (TRI) 利用
ROS 系統，完成將
SLAM 放置 Toyota's
Human Support Robot
(HSR)



NASA

利用 ROS 系統 開發
Robonaut 2，一款幫助
人類進入太空的機器人。

使用ROS系統開發的廠商

部分全球知名的廠商與機構，已開始使用 ROS 系統開發機器人

使用 ROS 系統開發的廠商



ROS Industrial consortium



工業機器人作業系統聯盟，是一個橫跨三大洲，擁有50個以上的世界性組織。

ROS Industrial consortium 成員



ROS-INDUSTRIAL

ROS-Industrial is an open source project that extends the advanced capabilities of the **Robot Operating System (ROS)** software to manufacturing.

Upcoming Events



ROS-Industrial Training (EU)

Oct 8, 2018 – Oct 12, 2018



ROS-I at The Assembly Show

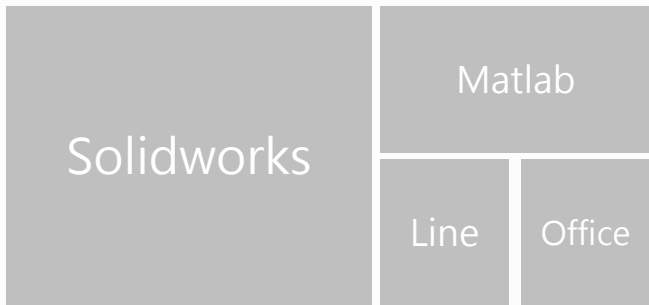
Oct 23, 2018 – Oct 25, 2018



ROS-Industrial EU Fall

ROS系統 與一般作業系統比較

ROS對機器人的硬體進行了封裝



作業系統
(Windows · Linux · IOS)

顯卡

音效

網卡

硬碟

攝影
機



機器人操作系統
(Robot Operating System)

雷達

攝影
機

伺服
馬達

深度
相機

加速
度感
測器

支援 ROS 系統的硬體設備

雷達

	Hokuyo URG-04LX- UG01	Hokuyo URG- 04LX	Hokuyo UTM- 30LX	Velodyne VLP-16	Velodyne VLP-64	SICK TiM series	SICK TiM series	RPLidar A1/A2
USB	USB2.0	USB 2.0	USB 2.0			USB 2.0	USB 2.0	USB 2.0
Serial		RS232				RS232	RS232 & 422	3.3V-TTL
Ethernet			100BAS E-TX	standard	standard	standard	standard	
Power Supply	USB 2.0	USB 3.0	External 12V/1A	External 9 – 32V	External 12 – 36V	External 9 – 28V	External 24V	USB 2.0
Power Highest	2.5W	4W	12W	8W	60W	2.2 ~ 4W	22W	7W
Figure								

支援 ROS 系統的硬體設備

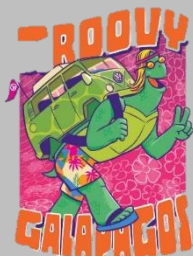
深度相機

	RealSense R200	RealSense ZR300 /Euclid	Orbbec Astra Pro	DUO-MC	Basler ToF	Kinect 2.0	ZED Stereo
USB	USB 3.0	USB 3.0	USB 2.0	USB 2.0		USB 3.0	USB 2.0/3.0
Ethernet					GigE (Vision)		
Power Supply	USB 3.0	USB 3.0	USB 2.0	USB 2.0	External 24V	External	USB
Power Consumption	Full: 1.6W	Full: 1.9W	Full: 1.9W	Full: 2.5W	Full: 24W	Full: 17W	Full: 2W
System Requirements	64-bit	64-bit				64-bit	
Figure							

支援 ROS 系統的硬體設備

一般相機

	Matric Vision	Basler AG	Occam Vision Group	Logitech	ADLink	National Instruments	Sony
USB 2.0	X*			X	X		X*
USB 3.0	X*	X	X	X	X	X	X*
GigE	X	X			X	X	X
Camera Link	X	X				X	X
FireWire		X				X	X
Figure							



2012
ROS - **G**roovy

2013
ROS - **H**ydro



2014
ROS - **I**ndigo

2015
ROS - **J**ade



2016
ROS - **K**inetic



ROS系統版本

ROS 從 2007 年問世開始，至今已邁入第 11 年。
而從 近幾年開始，ROS 基本上每隔一年便會有一個新版本出現。
並且版本代號的順序，是採用英文字母的順序。

ROS系統版本的選擇

Distro	Release date	Poster	Tuturtle, turtle in tutorial	EOL date
ROS Melodic Morenia	May, 2018	TBD	TBD	May, 2023
ROS Lunar Loggerhead	May 23rd, 2017			May, 2019
ROS Kinetic Kame (Recommended)	May 23rd, 2016			April, 2021 (Xenial EOL)
ROS Jade Turtle	May 23rd, 2015			May, 2017
ROS Indigo Igloo	July 22nd, 2014			April, 2019 (Trusty EOL)

ROS系統應用 Package

Client Layer

roscpp**rospy****roslisp****rosjava****roslibjs**

⋮

Robotics Application

Moveit!**navigation****Teleop
pkgs****descartes****mapviz****people**

⋮

Robotics Application Framework

tf**Robot_model****Dynamic
reconfigure****mavros****calibration****Vision
opencv**

⋮

Communication Layer

**Common
msgs****rostopic****rosmaster****rosparam****roslaunch****rosout**

⋮

Hardware Interface Layer

**Camera
driver****Joystick
driver****rosserial****Audio
common****diagnostics****3D sensor
drivers**

⋮

Simulation

**Gazebo ros
package****Stage ros**

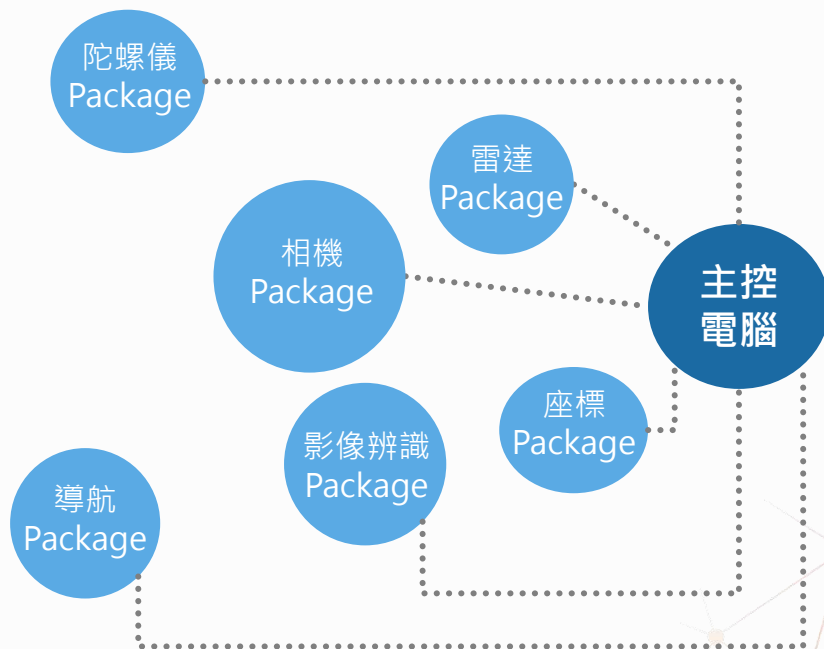
⋮

Software Development Tools

Rviz**qrt****rospack****roscdep****catkin****wstool**

ROS系統的設計思考

分散式控制系統



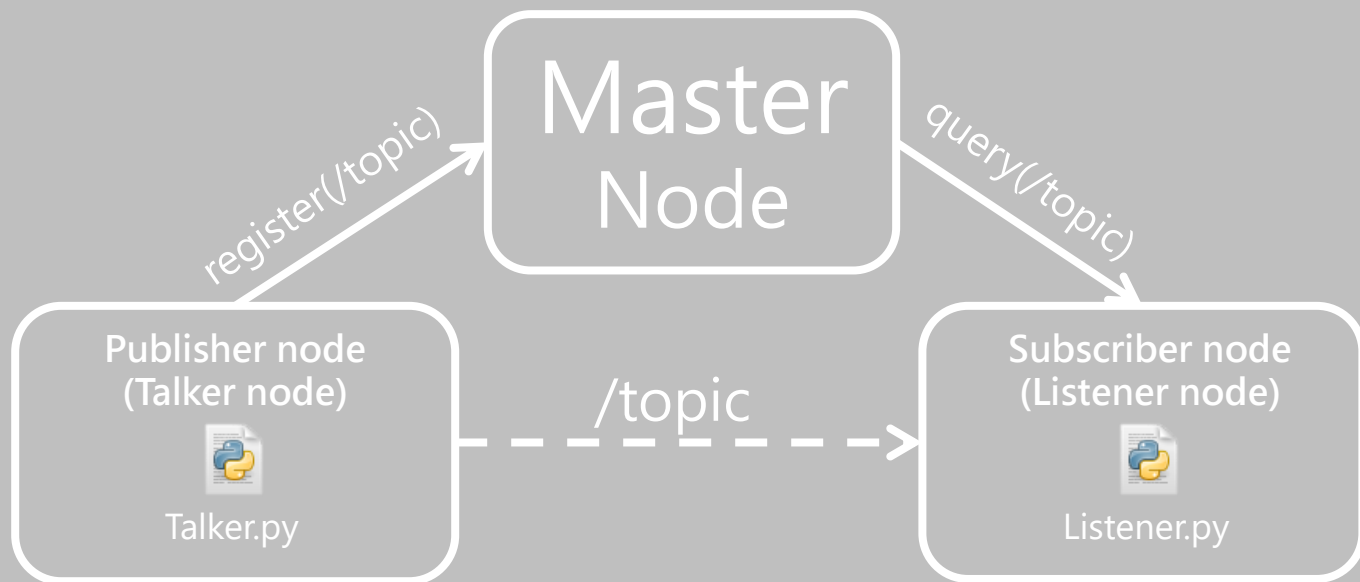
需要什麼，再抓什麼

每個人想要發展的機器人都不一樣，可能有人想要發展自駕車，需要雷達、影像裝置、GPS裝置等等...；也有人想要完成倉儲機器人，也需要雷達、影像裝置等等....但是卻不需要GPS裝置。

因此，若是想要讓全世界的人都使用ROS系統開發機器人，就必須要將全世界各種種類的感測器匯入系統中，但這顯然是不現實的想法。所以ROS系統便採用分散式控制系統的概念，讓開發者自行選取想要的感測器，再去抓取相對應的Package即可。

ROS系統的組織結構

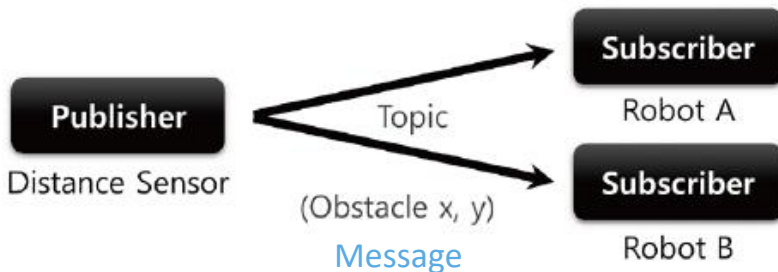
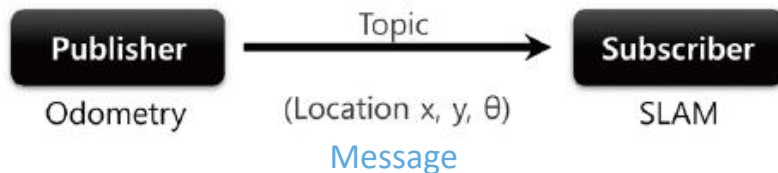
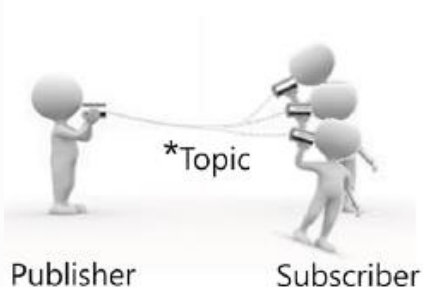
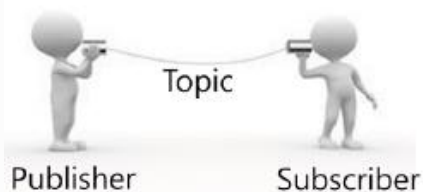
Node and Topic



備註：節點不限定使用 Python 撰寫

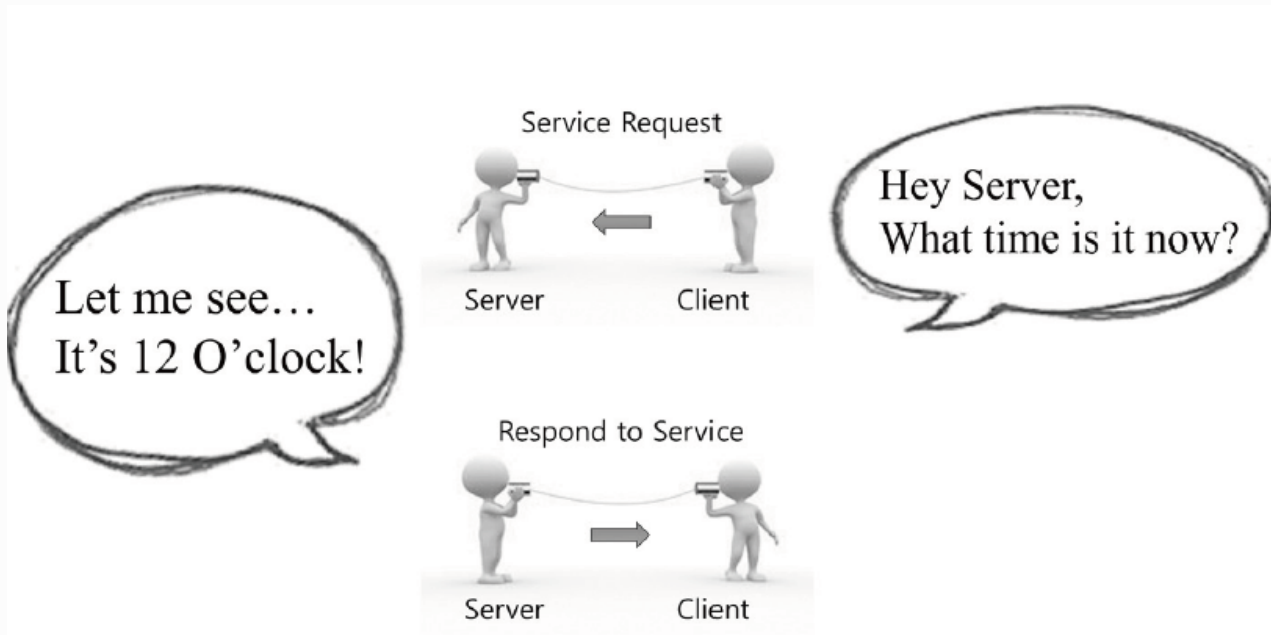
ROS系統的組織結構

Topic 的傳遞 與 Message



ROS系統的組織結構

Service 的傳遞



ROS系統的組織結構

Package 與 Node 的關係



Package_1



node1.py



node2.py



node3.py



node4.py



Package_2



node5.py



node6.py



node7.py



node8.py



Package_2



node9.py



node10.py



node11.py



node12.py

ROS系統的組織結構

ROS系統的文件設置

安裝目錄



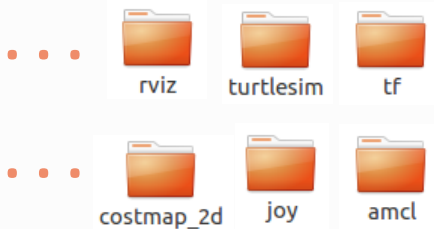
ros



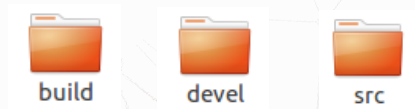
catkin_ws

工作空間

ROS 的 軟體工具
存放使用 apt 套件安裝的 package



ROS 專案需使用到的空間
存放 build 、devel 、src



ROS系統的組織結構

ROS系統的工作空間



工作空間



利用 Catkin 系統建立工作空間是需要的文件檔案



存放該工作空間所需要的msg、srv、packag需要的程式庫等文件



存放該工作空間所需要的 package 與 Node

在多台機器上運行ROS系統

以筆電與樹梅派為例子



為什麼選擇 ROS ?

ROS 有良好的機器人開發環境



開源的系統

ROS 使用了 BSD 的授權條款，讓使用 ROS 的開發者除了研究、教育等用途外，也能夠進行商業化的用途。



支持多種程式語言

考量到眾多人員習慣開發的程式語言不同，ROS 便支持了眾多主流的開發語言，如 C++、Python、Java等程式語言。



可視化工具的使用

ROS 擁有的可視化工具，如 rqt、rviz 等工具，讓機器人開發人員能夠更快速、更直覺地調校機器人，讓開發時程可再縮短。



統一的資料格式

「書同文、車同軌」便是最好解釋 ROS 對於機器人研究快速發展的一個代名詞。這讓使用 ROS 開發機器人的開發者，不論使用何種程式語言，都可以互相的溝通有無。讓開發人員能夠快速地使用前人所開發的機器人技術。



方便的通訊架構

ROS 通信架構，便是 ROS 整個框架的一個基礎。這讓「進程間通訊」變得簡易許多。並且使用 TCP / IP 的架構，讓不同機器間的通訊可以較為快速地進行溝通。



廣大社群使用者

ROS 最遠大的目標，便是讓世界上所有人都使用 ROS 開發機器人。截止 2018 年為止，也已經擁有超過 55 家以上的世界級企業採用 ROS 系統作為研究與開發機器人的系統。而每年的社群指標各項指數也都快速地增長中。

Thanks for coming

Have a nice day!