

# Table of Contents

1. ROS Introduction .....	2
1.1. An introduction.....	2
1.2. Nodes .....	3
1.3. Topics.....	3
1.3.1. Subscribing.....	3
1.3.2. Publishing.....	4
2. ROS Tutorials.....	4

## Welcome

### Project Willy

- [History of Willy](#)
- [Project Willy](#)
- [Publicity](#)
- [Sponsors](#)

### Getting started

- [Introduction to ROS](#)
- [Development Guide](#)
- [Driving Willy](#)
- [Manual](#)
- [Wiki Manual](#)

### Build of Willy

- [Design history](#)
- [Hardware](#)

### Architecture

- [Software Architecture](#)
- [ROS topic design](#)

### Raspberry Pi's

- [Sensor node](#)
- [Social Interaction node](#)
- [Power node](#)

### Components

- [ROS master](#)
- [New ROS master on Lubuntu](#)
- [Sonar](#)

- [Lidar](#)
- [Kinect](#)
- [Localization and navigation](#)
- [Motor controller](#)
- [Joystick](#)

## Lessons learned

- [Todo & Advice](#)
- [Lessons Learned](#)

## Archive

- [Previous Groups](#)
- [Research Archive](#)
- [Skylab Architecture](#)
- [Skylab](#)
- [Multi master](#)
- [WillyWRT](#)
- [Realisation](#)
- [Hardware](#)
- [Brain](#)
- [Design Guild](#)
- [Social interaction](#)
- [Speech](#)
- [Speech recognition](#)
- [IMU](#)
- [Human Detection](#)
- [Radeffect App](#)

# 1. ROS Introduction

## 1.1. An introduction

As a requirement from the product owner, **ROS** is used as framework on Willy. ROS, the Robot Operating System, is a flexible software **framework** for use in robots. It consists of a collection of libraries, tools and conventions that provide basic infrastructure to communicate between different parts of the robot.

In the case of Willy, ROS is especially handy because Willy is made with a modular design. All modules can be removed without disrupting the other functionalities of Willy. For example, when

the web interface is removed, Willy is still able to drive, but with another module as for example the keyboard controller. Or the removal of the motor driver makes Willy still able to interact with public.

## 1.2. Nodes

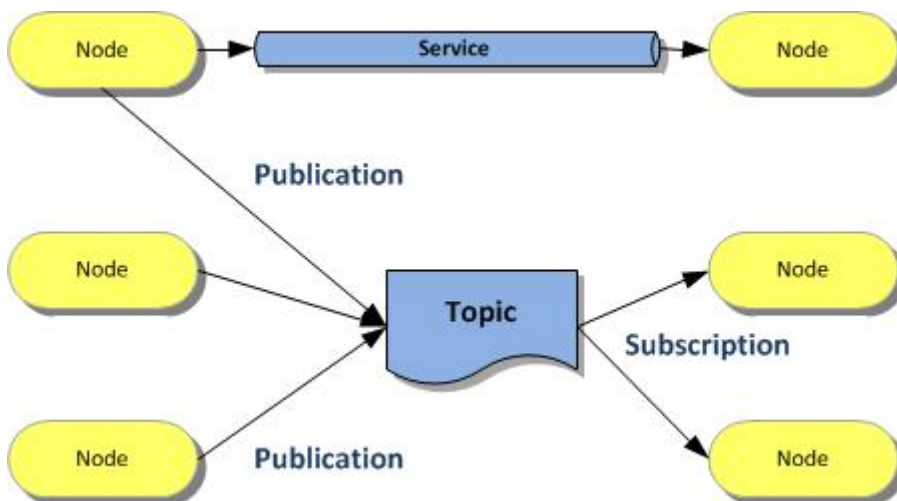
A **node** in ROS can be seen as a module. It is an executable that communicates through ROS to other nodes to send and receive data. A node can be for example a c++ application, or a piece of Python code, or even an Arduino connected with USB running code. A piece of information a node receives or sends is called a **message**.



More information can be found at <http://wiki.ros.org/ROS/Tutorials/UnderstandingNodes>

## 1.3. Topics

A topic is a bus over which nodes can exchange data messages. A topic always has a name so that all topics can be identified clearly.



More information can be found at <http://wiki.ros.org/Topics>

To interact with a topic, two methods are used, subscribing and publishing.

### 1.3.1. Subscribing

**Subscribing** is getting data from a topic. This does require the topic to be published first. Each time a message is published on this topic, a message will be passed to all subscribing nodes containing the data within. This way a node can use this information.

For example, the move\_base node will publish a message on the topic `/cmd_vel` with acceleration values. The motor controller is subscribed to this topic and will thus read the acceleration values, letting the robot move appropriately.

### 1.3.2. Publishing

**Publishing** is sending data to a topic. Upon starting a node, it will create the associated topics and start publishing information upon these topics. Whenever a node has new information, a message will automatically be sent to the linked topic(s), upon which the subscribers will receive and process the message.

## 2. ROS Tutorials

We found out the hard way, multiple times, that ROS can be quite difficult to comprehend. The principle seems rather easy, have a working network, communicate on TCP-IP and DNS name, and start coding. However, there are a variety of pitfalls where you can go wrong.

We have used multiple ROS tutorials to understand more of the technical aspects of ROS as well as the (im)possibilities it holds. Due to these tutorials we kept Willy as it was, but also gained knowledge about ROS. Therefore we strongly suggest to read and try some of the tutorials ROS provides on their webpage.

<http://wiki.ros.org/ROS/Tutorials>