ROS Periphery Interface Module
Integrator manual

Project title	Digital Technologies, Advanced Robotics and increased Cyber-security for Agile Production in Future European Manufacturing Ecosystems
Project Abbreviation	TRINITY
Project Funding Scheme	H2020 Innovation Action (IA)
Call Identifier	DT-ICT-02-2018: Robotics - Digital Innovation Hubs (DIH)
Project Website	http://www.trinityrobotics.eu/
Project Start Date	1.1.2019
Authors	JSI

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 825196.

Abstract

The ROS (Robot Operating System) periphery interface enables fast swapping of exchangeable peripheral elements in a robotic workcell. It ensures ROS communication between the peripheral elements and the ROS backbone of the workcell. It acts as a proxy between the ROS-based workcell and exchangeable elements of the cell without ROS capabilities. This manual presents the components of the module, installation procedure, and example implementations of standard exchangeable modules. Specific software instructions are given. Instructions for a general hardware components are hard to give, as the peripheral elements can vary substantially. Instead, a general overview is provided with two example implementations presented in Section 4.

Contents

1 Overview

In order to increase the flexibility of production, the robot workcell needs to enable quick reconfiguration. This can be achieved by exchanging peripheral elements of the cell. These elements can include relatively small modules like robot end-effector tools, or larger modules that are plugged to the backbone of the workcell (e.g. plug-and-produce trolleys). While some elements can be designed as stand alone, some might not include a ROS enabled computer and therefore lack ROS capability. The ROS periphery interface enables swapping of these elements by providing a bridge between them and the cell's ROS system. A simple overview of a ROS supported workcell with a ROS periphery interface is shown in Fig. 1.

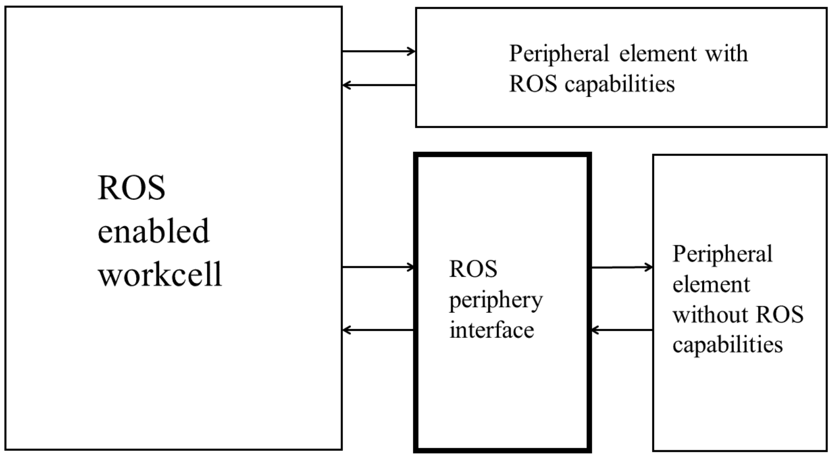

Figure 1: Overview of the cell with the ROS periphery interface.

1.1 Robot Operating System (ROS)

To achieve the desired software modularity, we must enable the data flow between different workcell modules. The data needs to be structured in such a way that it can be parsed by all modules within the system. With this is mind, we designed a software infrastructure for robot workcells based on Robot Operating System (ROS) framework [1]. ROS provides the desired software flexibility. While many features and tools are included in ROS, a subset of them is used to realize conectivity between the modules of the cell: ROS nodes, topics, services, action and parameters servers, messages, etc.

The integration of newly developed modules and peripheral elements into the workcell can be accomplished with ease in a ROS-based system. When the functionality of the cell needs to be enhanced by including new modules, it is not necessary to change the existing ROS communication components (i.e. ROS nodes). It is sufficient to add the desired functionality to the new component by developing just a new ROS node without changing the existing ROS nodes. While some hardware modules already include a computer and thus enable easy integration into a ROS-based software environment, others do not include a computer and thus cannot be easily integrated into ROS.

1.1.1 ROS installation

The ROS periphery interface is based on the Kinetic version of ROS. It can be installed on a micro-computer running a Linux Ubuntu 16.04 LTS operating system. The tutorial covering the installation of Ubuntu 16.04 can be found at this link. Alternatively, the current version of Raspbian OS can be used if Raspberry Pi is used as micro-computer to implement the interface.

Once Ubuntu 16.04 is installed and running, ROS Kinetic can be installed by following instructions found at this link.