Open Source Software for Robotics

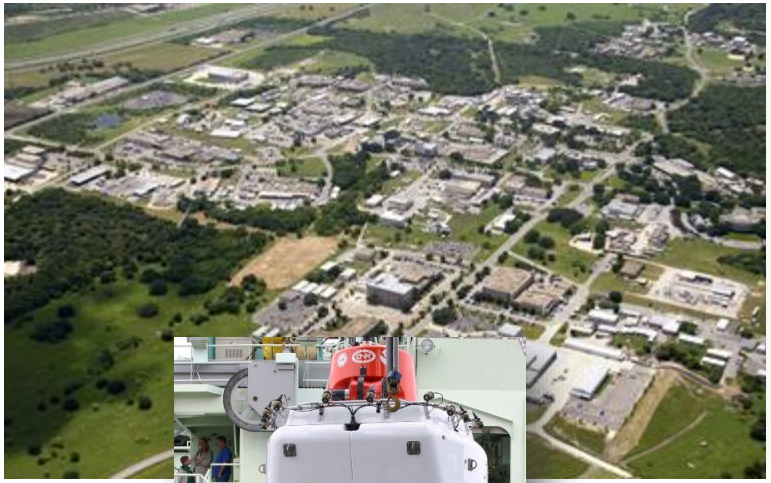
When to use it and when to use closed source

Matt Robinson

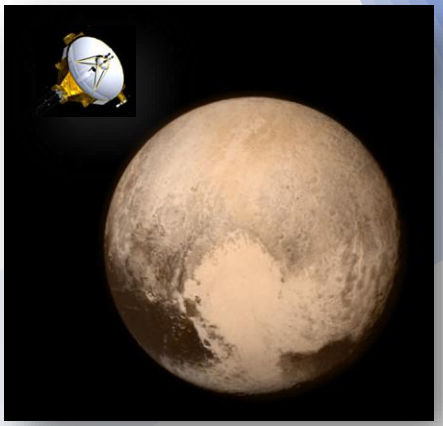
Southwest Research Institute

March 16, 2022

SwRI: Deep Sea to Deep Space



Alvin submersible



New Horizons, Pluto

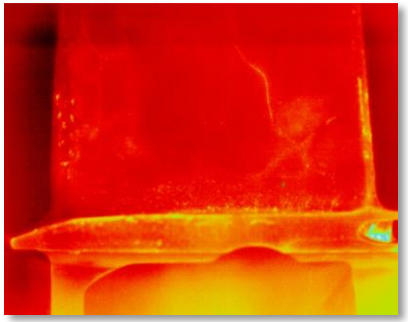
Southwest Research Institute Characteristics

- Est. 1947
- San Antonio, Texas, USA
- Independent, Not for profit
- Applied RDT&E Services
- Natural Science and Eng.
- FY 2020 Revenue: $696M

ROS industrial consortium americas

SwRI

Manufacturing Robotics & Technologies



Machine Vision and Perception

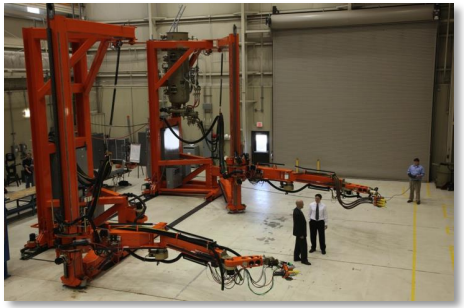
robotics.swri.org

Industrial Automation and Controls

ROS industrial

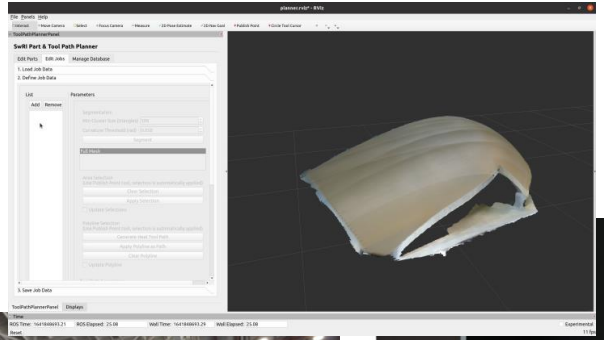
Advanced Robotic Software

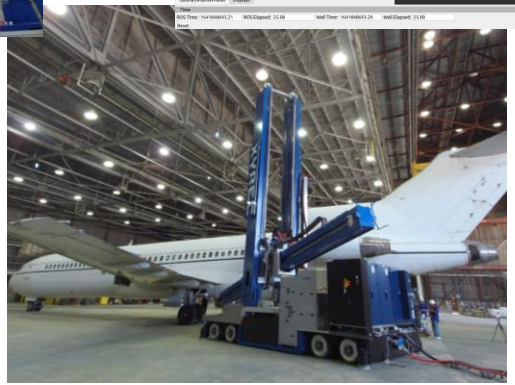
Custom Robotics



System Integration

SwRI Advanced Systems Development





Reference: robotics.swri.org

Let’s understand a few things…

- What Is Open Source Software
- When would you choose open source?
- Open Source Software use for commercial purposes
- Common Open Source licenses to be aware of
- Open Source Software and vulnerabilities
- Things to consider in leverage of Open Source Software

What kind of open things are there?

- Standards
  - A standard that is freely available for adoption, implementation and updates
    - XML
    - SQL
    - OPC-UA
    - MTConnect
  - Allow value to be brought for business collaborators by agreeing to rules/updates etc to maintain interoperability and quality
- Software
  - Libraries or combination of lines of code that are made available, as source code, that is freely available
  - *Most* licenses allow for redistribution and modification by anyone, anywhere with attribution.
  - Some license type will dictate any updates from contributors will also become free and open – *the devil in the details*
- Frameworks

When do you choose OSS?

- You cannot find a solution that meets your requirements
  - Level of autonomy
  - Level of efficient human and robot interaction
  - Mix of parts/processes and/or mobility
- You find yourself continuously having to rewrite applications across environments
- You have needs for pieces that are not key to your IP position, aka “secret sauce”, but enable your group/team to innovate
  - Leverage the community to get building blocks
  - Enable a bigger “team” to means test components
  - Enable more efficient transition from academia to shop floor

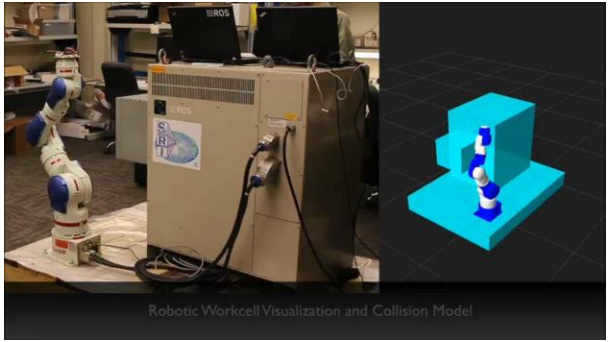
How did the idea of OSS in robotics come to SwRI?

- Long history of building unique solutions
- Often rewriting software based on hardware changes
- 2010 via Unmanned Ground Systems work discovered ROS
  - Navigation Utilities
  - Abstracts away hardware
  - Enables efficient iteration on hardware

Where did the idea of OSS in robotics come to SwRI?

- 2011: ROS-Industrial Inception
- 2012: ROS-I Repo Launch
- 2013: ROS-I Consortium Launch

Shaun Edwards: Founder Paul Hvass: Consortium PM




```
commit e9494d21fbe5a6d951e7534f1c9c088236dcad69
Author: sedwards@willowgarage.com <sedwards@willowgarage.com@076cdf4d-ed99-c8c1-5d01-9bc0d27e81bd>
Date:   Fri Sep 23 02:12:56 2011 +0000

    Check in of initial directory structure.  sia20d robot urdf commited with primatives.
```

A Disruption in Software for Automation

Enter ROS – Robot Operating System

- Open Source
- Established to prevent re-inventing the wheel
- Maintained by Open Robotics
- Reusable Software Components
- >1,000,000 user downloads/mo

ROS

is…

A **Middleware** Framework

An International **Open-source** Project

A Library of **Free Software and Tools** for Robotic Development

ROS Releases and Journey to Industry

2008

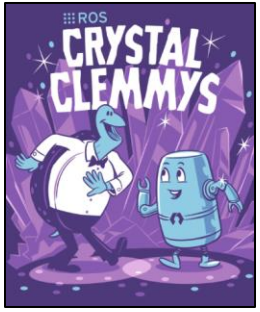
- PR2 and ROS start at a research platform for universities and research institutes

Jan 2010

- ROS 1.0 is released with tutorials
- 12 releases between 2010-2018

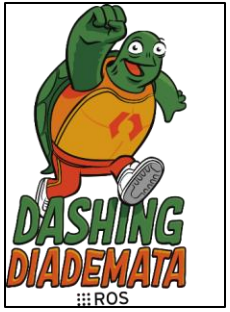
Dec 2017

- First Beta release of ROS 2.0 for general use



Dec 2018

- Actions support
- Navigation package



May 2019

- Multi-axis robot motion planning

Jun 2021

- Latest release

10 Year Development Cycle

ROS 2.0 Industrial Use

Start using for next generation platform development

So about those licenses

- Two Main Groups
  - Permissive - guarantees the freedom to use, modify, and redistribute, while also permitting proprietary derivative works
  - Copyleft - developers have the right to use, modify, and share the work as long as the reciprocity of the obligation is maintained. Using a component with this kind of open source license requires that you too must make the code open for use by others
- Devil is in the details
  - Can you claim IP based on the open source you utilized? Do you have to commit open source back should you commercialize?
- Some help… GitHub's choosealicense.com site,
  - Created to help developers working on open source projects easily find the open source license that suits their needs
  - Offers an appendix that compares open source licenses maping out differences based on what is permitted/restricted

Building Commercial Solutions…

- It is a thing happening all the time!
  - https://www.techrepublic.com/article/the-hottest-thing-in-robotics-is-an-open-source-project-youve-never-heard-of/
  - https://www.automate.org/industry-insights/ros-industrial-for-real-world-solutions

Melonee Wise, Robot Ninja / CEO of Fetch Robotics
Answered 2 years ago

Why is ROS not good for industry?

It's important to recognize that "ROS is not good for industry" is a blanket statement that is very non-specific.

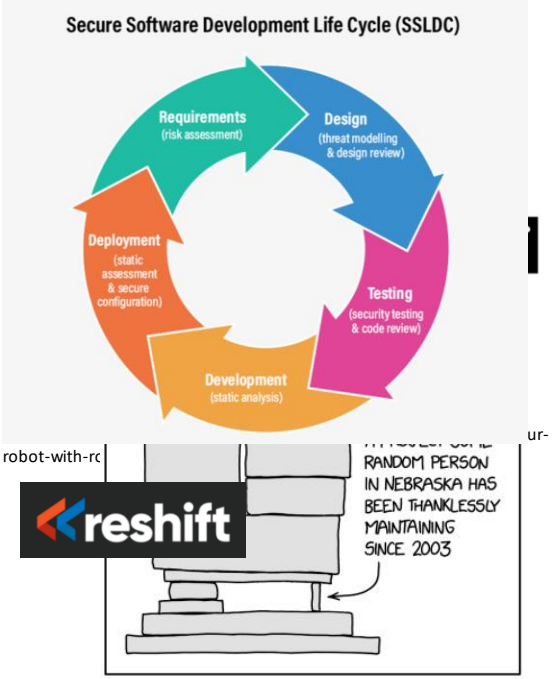
Let's dive into many of the common complaints:

1. Can't be used in real-time systems - "real-time" means something very specific, which is fixed timestep processing that can't be blocked other processes. Some systems may or may not need real-time and that real-time doesn't need to be handled at the OS/ROS level. For example, the real-time process can happen on an embedded system and the communication to that system does not need to be real-time. So for industrial applications that don't need real-time processes at the OS/ROS level, ROS is completely fine.
2. Limited embedded support - This again only matters for industrial applications that need ROS to run on an embedded system and ROS does have embedded support for some hardware. It just depends on whether that hardware is appropriate for the application.
3. Not secure - Yes, ROS is not secure but there are workarounds for making it secure using VPN to encrypt the communications.
4. No multi-robot support - The ROS master architecture does have some limitations for supporting multi-robot applications but again there are workarounds.

In general, I would say that ROS does have some limitations but those limitations haven't prevented some companies, like Fetch Robotics, from building successful companies and products with ROS.

Open Source and Security

- There are layers
  - The code itself
    - Secure Software Development Life Cycle Processes (SSDLC)
    - Improve existing code by addressing security issues
      - Use compilers/interpreters tools, use of LGTM (GitHub), Coverity, Reshift
    - Awareness and Reporting of Common Vulnerabilities and Exposures
    - Explore Dependencies – SCA Tools, GitHub's Dependency Graph
    - ROS 2 and DDS – Cryptography, Authentication, Access Control, Logging
      - Sros2 – enables configuration of security in ROS 2
  - How the application is built & Deployed
    - Secure Connections – connect robots to separate network segments, disable USB usage
    - OS Hardening – do not have "admin" users, local users least privileges, close unused ports, patch available security updates
    - National Institute of Standards and Technology (NIST) Cybersecurity Framework & International Society of Automation (ISA)



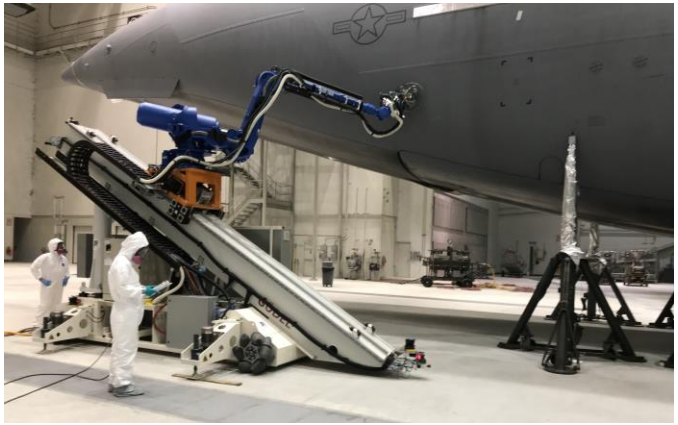
Credit: Florencia Cabral Berenfus – Canonical – ROS-I Community Meeting – Cybersecurity Challenge

And what about frameworks?

- ROS is not just libraries or open source software but also a framework
- Frameworks provide a combination of utilities, libraries, integrated in a way with documented interfaces to enable reuse for new or different applications
- Examples
  - A5 and the complimenting ARM Institute project – Autonomous Coatings
  - ConnTact Assembly Framework sponsored by NIST

Leveraging an Open Framework

- Recent success taking a developed application framework (A5) and extending it to new platforms and processes
  - ARM Institute Sponsored
    - https://arminstitute.org/project-highlight-mobile-autonomous-coating-application-for-aircraft-sustainment/
    - https://arminstitute.org/project-highlight-autonomous-coating-with-realtime-control-and-inspection/
- Demonstrated adaptability to new hardware and improved existing tools and developed new capabilities

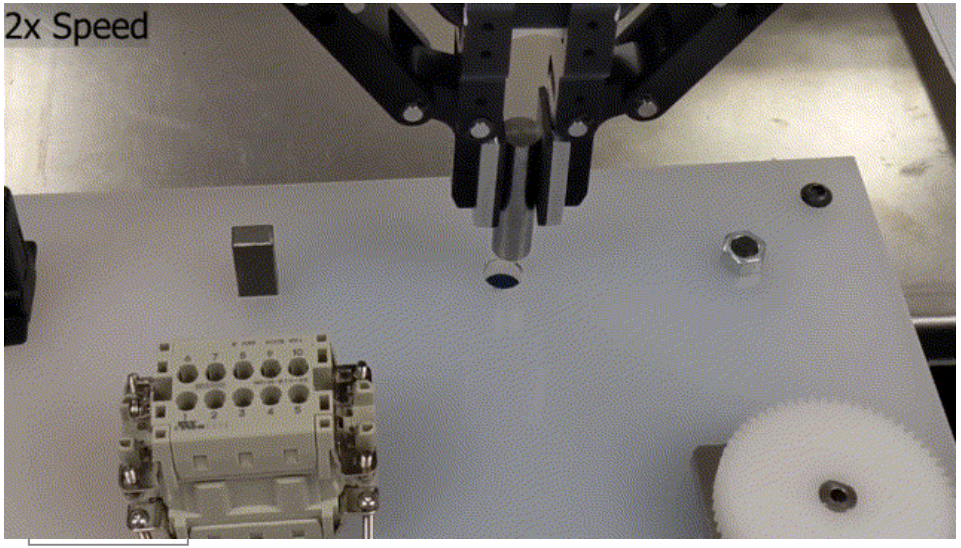


Extensibility Result

https://arminstitute.org/project-highlight-mobile-autonomous-coating-application-for-aircraft-sustainment/

Agility in advanced assembly applications

- The ConnTact Assembly Framework
  - Ability to enable researches to simply implement and test learning algorithms to test extensibility
  - Supported by NIST and the Agility Working Group



https://github.com/swri-robotics/ConnTact

Why would a business consider open source?

- A quick place to start
- Students are graduating familiar with a number of robotics open source tools
  - OpenCV
  - Point Cloud Library
  - ROS
  - MoveIt
- A means to leverage research in a modular fashion
  - Prototype and switch to target hardware later
  - Continuous improvement
  - Able to focus just on IP component on top of existing tools
  - Accelerate Tech Transition

Your business wants to take advantage of OSS

- Develop an open source strategy
  - What do you want to do?
  - What are your objectives?
  - Can/how you participate?

  } Different considerations if for "in-house" use or making a product to sell
- Become a part of a community
- Become familiar with licenses
- Consider if you consume offering improvements back to projects
  - Submit PRs/Fixes
  - Offer feature requests and specific feedback on desired behavior
  - Support REPs and/or Working Groups
- Don't fork something if you do not have to
- Do not open source if you cannot maintain what you open source

Accelerating Product to Market

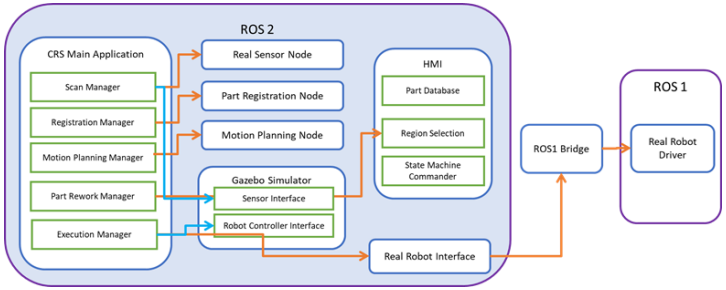
- Enabling foundation
- The right partners
- Plan – concept through sustainment

Source: https://github.com/swri-robotics/euler

Basition Solutions, ProMat 2017

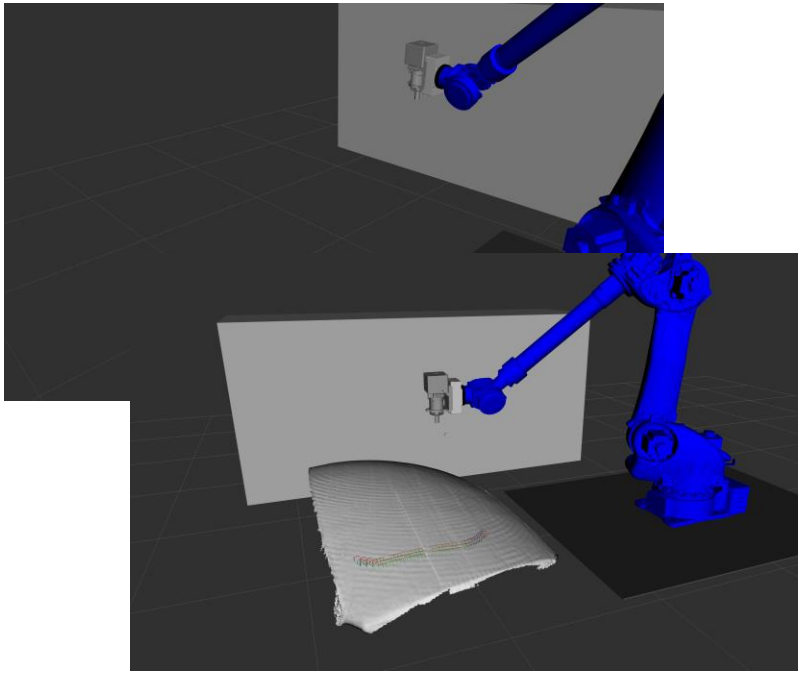
First Industrial ROS 2 Application

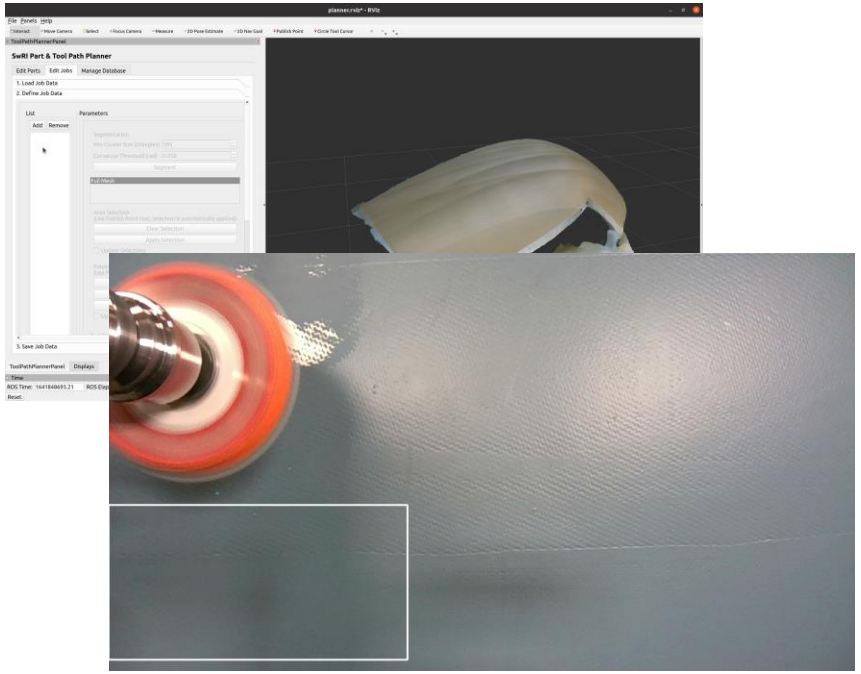
- Advanced Robotics for Manufacturing Institute – Collaborative Robotic Sanding of Aircraft Panels
  - Prime – Spirit AeroSystems
  - Fully open-source application
    - Stands as example
    - Make modular for re-use
    - Drop in capabilities



https://github.com/swri-robotics/collaborative-robotic-sanding

Adaptive Surface Finishing Framework

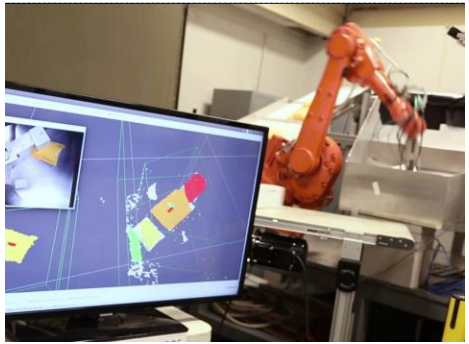


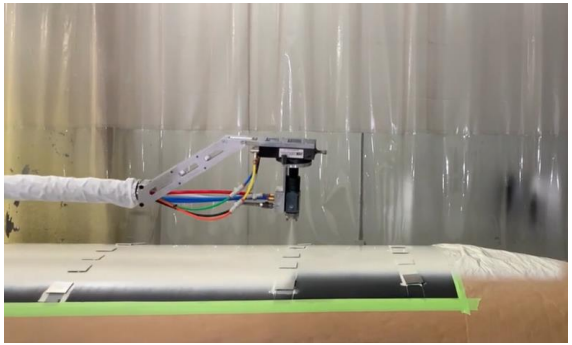


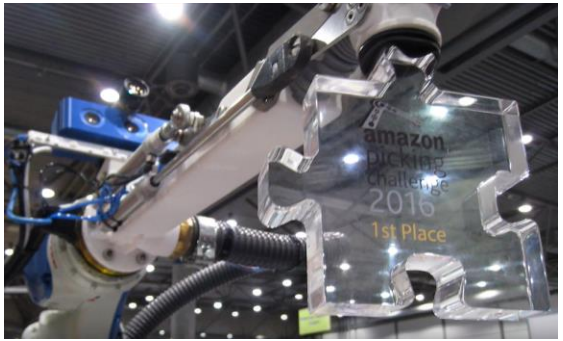
Resources for the Community

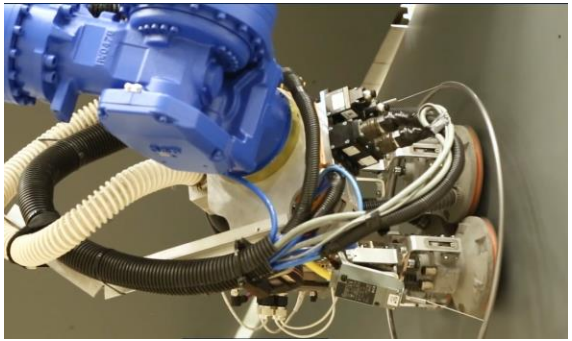
- ROS-Industrial
  - Home: rosindustrial.org
  - Documentation: wiki.ros.org/industrial
  - Code: https://github.com/ros-industrial; https://github.com/ros-industrial-consortium
  - Training: http://ros-industrial.github.io/industrial_training/
  - ROSin: http://rosin-project.eu/
- Upcoming Events (https://rosindustrial.org/events-summary/)
- SwRI Robotics: https://robotics.swri.org, https://github.com/swri-robotics

Impact: Accelerating Advanced Robotics









Thank You!

Matt Robinson

ROS-Industrial Consortium – Americas

Program Manager

matt.robinson@swri.org

210-522-5823

rosindustrial.org

robotics.swri.org

https://rosindustrial.org/events/2022/6/10/ros-industrial-consortium-americas-2022-annual-meeting