OSRF Projects Portfolio

November 2015

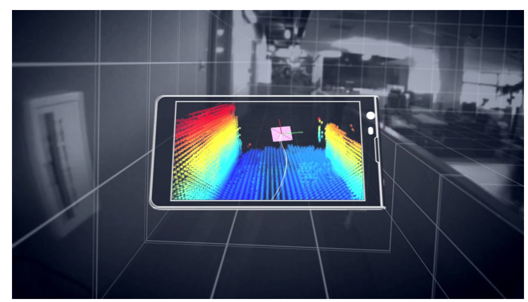
info@osrfoundation.org

Table of Contents

Project Tango

Project Tango: Overview (1 of 3)

Project Tango brings spatial perception to the Android device platform by adding advanced computer vision, image processing, and special vision sensors.

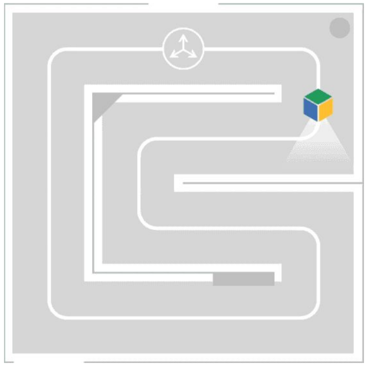


This allows mobile devices to navigate the physical world similar to how we do as humans.

Project Tango: Overview (2 of 3)

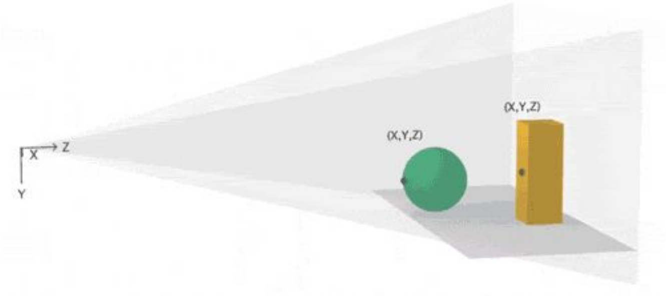
Motion Tracking

Allows a device to understand position and orientation using custom sensors.



Depth Perception

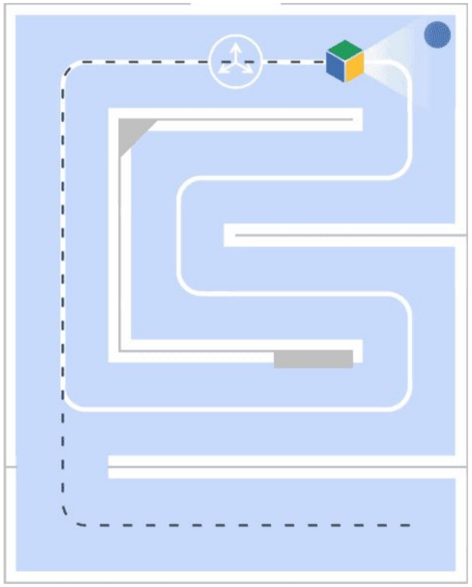
Depth sensors can tell you the shape of the world around you.



Project Tango: Overview (3 of 3)

Area Learning

Project Tango devices can use visual cues to help recognize the world around them. They can self-correct errors in motion tracking and relocalize in areas they've seen before.



Project Tango & OSRF

Project Description

Project Tango is a complex software project that handles large quantities of data. We helped Google adapt two tools from ROS to improve Tango.

Timeframe

May 2013 – Present

Funding Source

Google

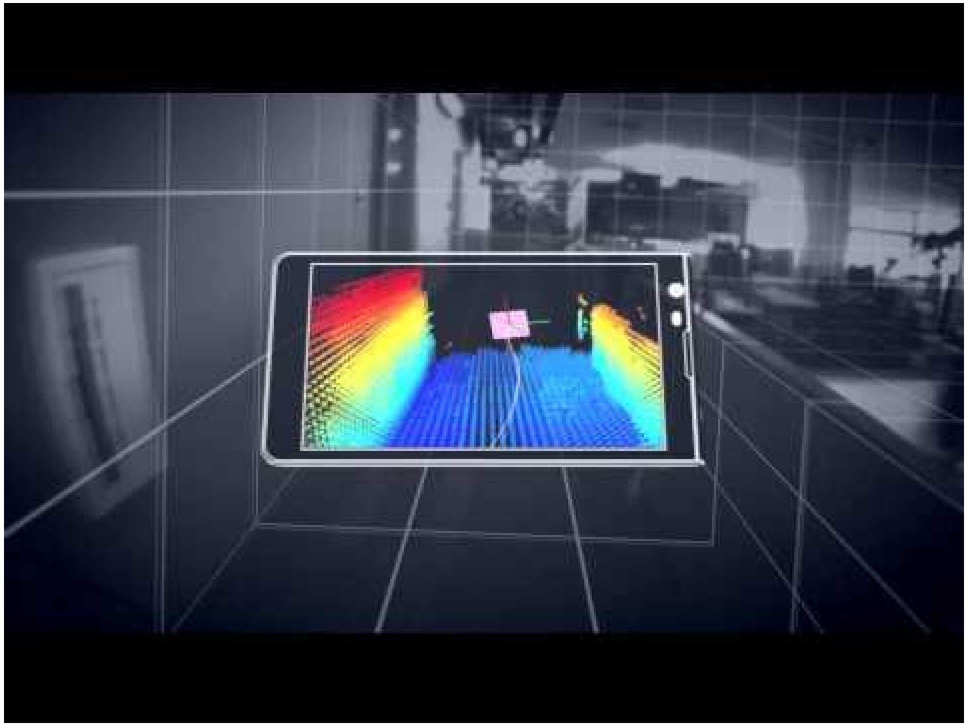
ROS

Project Tango: Goals

We helped Google adapt two tools from ROS to improve Tango:

- The ROS build system, which is designed to handle complex software projects
- The ROS logging system, which does high-performance data logging

Project Tango: Video



https://www.youtube.com/watch?v=Qe10ExwzCqk

Project Tango: Technical details

Technologies Used

Built on the ROS build system, catkin, and the tools for that system

Android as the target architecture

Used O_DIRECT style writing to maximize rosbag recording throughput

Used C++11 to replace use of boost in parts of the ROS system for use on Android

Project Tango: Technical contributions

Available to the ROS community

Improved build tools including more build performance and better UX over existing tools (e.g. `catkin_make_isolated`).
https://github.com/catkin/catkin_tools

Custom-built rosbag writing that optimizes the writing of rosbags for use on Tango devices and Android. The library can be used for a performance boost on PC's, too.
https://github.com/osrf/rosbag_direct_write

DARPA Robotics Challenge (DRC)

DRC: Overview (1 of 3)

A worldwide competition of robot systems and software teams vying to develop robots capable of assisting humans in responding to natural and man-made disasters.

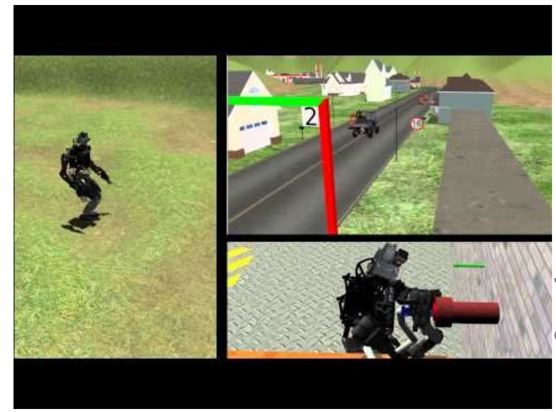
robohub.org

DRC: Overview (2 of 3)

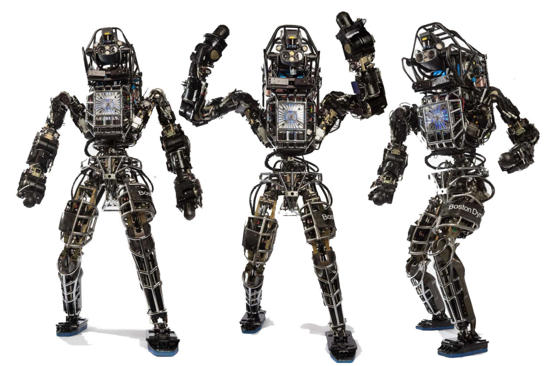
Teams competed to win an Atlas robot during the **Virtual Robotics Challenge** (VRC).

They completed three simulated tasks in Gazebo:

- Driving
- Terrain traverse
- Hose manipulation

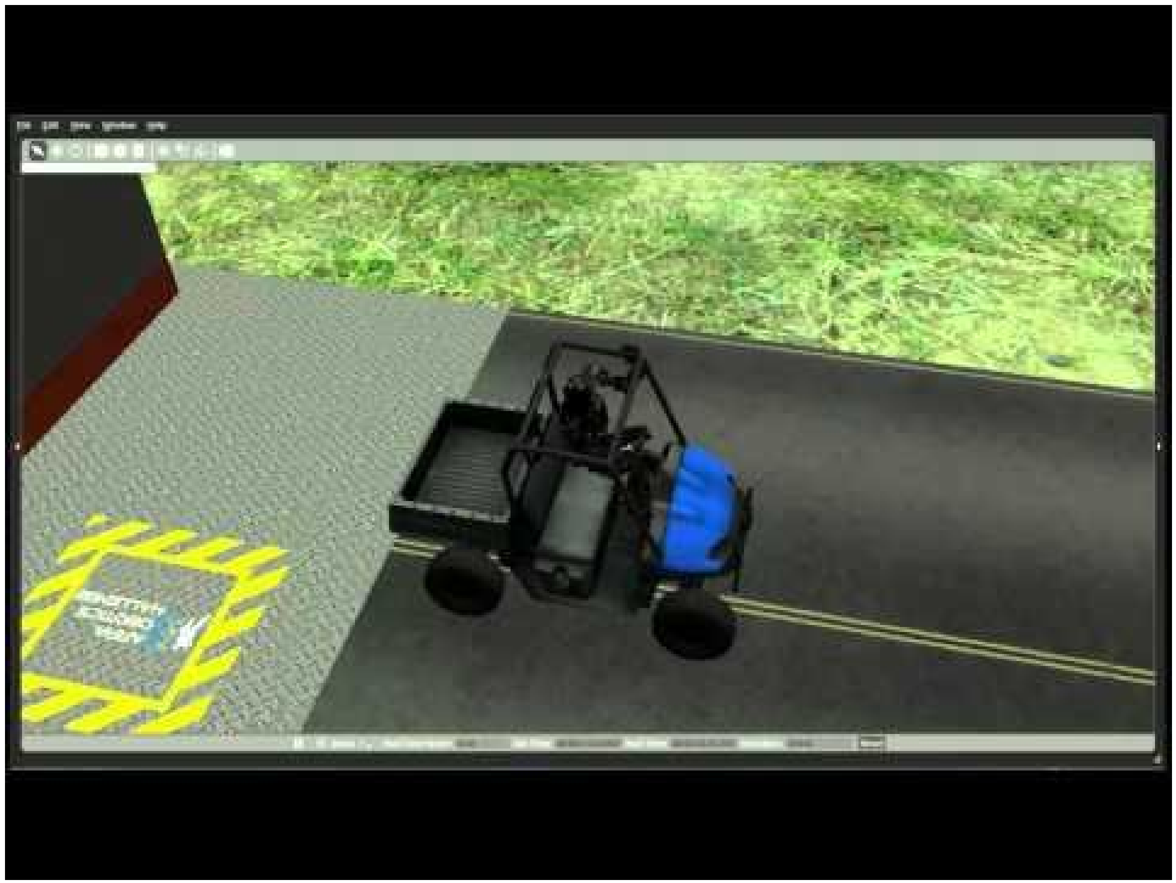


i1.ytimg.com



technobuffalo.com

Virtual Robotics Challenge: Video



https://www.youtube.com/watch?v=k2wVj0BbtVk

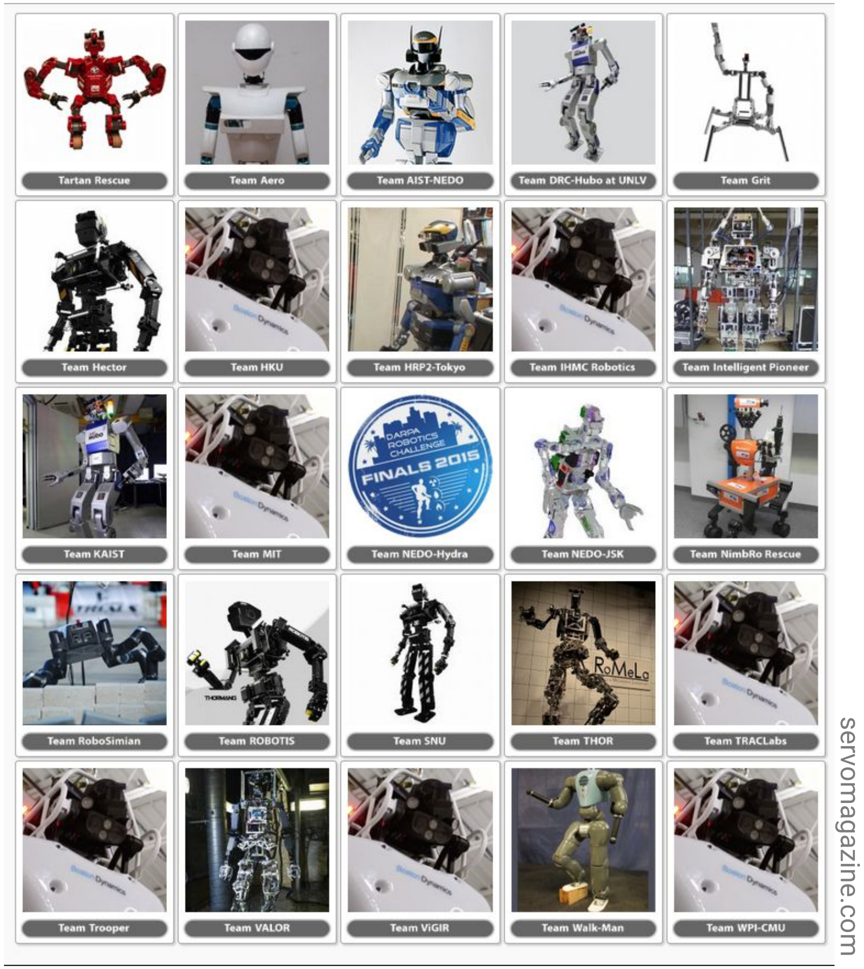
DRC: Overview (3 of 3)

Two subsequent physical competitions demonstrated teams' mobility, manipulation, dexterity, perception, and operator control mechanisms.

The top 3 teams won $2 million, $1 million, & $500k.

www.livescience.com www.roboticsselect.com

DRC



DRC Finals teams

DRC & OSRF

OSRF developed and improved upon Gazebo for use in the Virtual Robotics Challenge.

This effort involved performance improvements, creation of cloud resource interfaces, and support of complex simulation environments.

GAZEBO

Timeframe

Sept 2012 – Aug 2015

Funding Source

DARPA

DRC: Contributions (1 of 4)

Real-time robot simulation using cloud resources

CloudSim

Facilitated the use of four different physics engines in a single simulator

Open Dynamics Engine™

SimTK

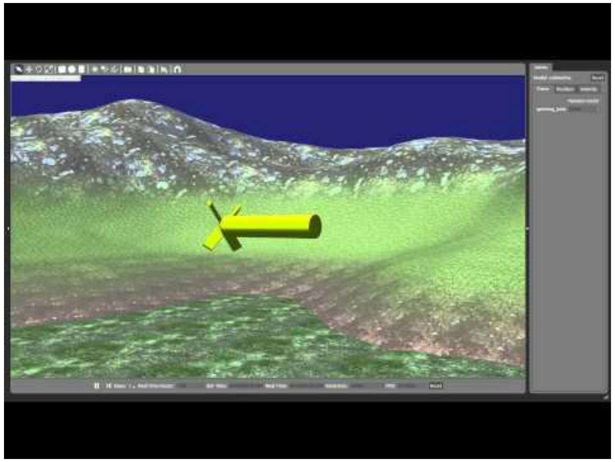
DART
Dynamic Animation and Robotics Toolkit

Bullet Physics Library

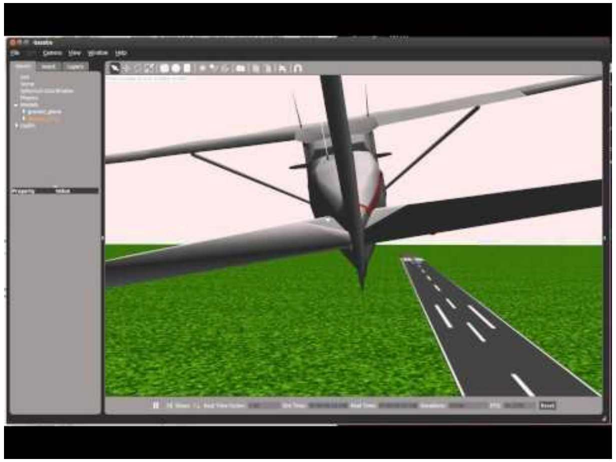
DRC: Contributions (2 of 4)

Hosted an online competition involving 22 teams distributed around the world

Integrated aerodynamics and hydrodynamics into simulation



https://www.youtube.com/watch?v=Jmz-N7zqK8g



https://www.youtube.com/watch?v=Jmz-N7zqK8g

DRC: Contributions (3 of 4)

Of the 23 teams in the DRC Finals:

18 teams ran ROS

14 teams used GAZEBO

DRC: Contributions (4 of 4)

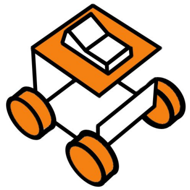
“The Virtual Robotics Challenge itself was also a great technical accomplishment, as we have now tested and provided an **open-source simulation platform** that has the potential to **catalyze the robotics and electro-mechanical systems industries** by lowering costs to create low-volume, highly complex systems.”

- Gill Pratt

Former DARPA Program Manager, DRC

MENTOR2

Manufacturing Experimentation and Outreach



MENTOR2: Overview

The MENTOR2 program seeks to create a learning environment for training students to build and repair electromechanical systems in austere environments.

These tools, which include software and manufacturing equipment, should support iterative rapid prototyping and testing in the field.

MENTOR2 & OSRF

Project Description

The Gazebo Design Kit (GDK) lets users construct and simulate electromechanical components.

Together with SRI's MOOC, the GDK forms the MENTOR2 learning environment.

GAZEBO

SRI International

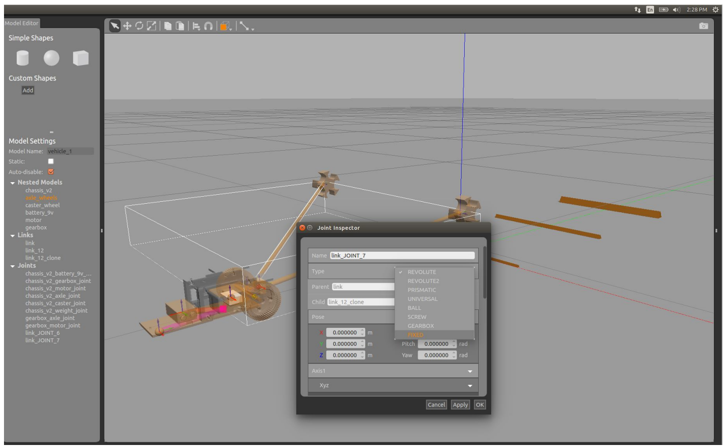
Timeframe

July 2014 – Present

Funding Source

DARPA

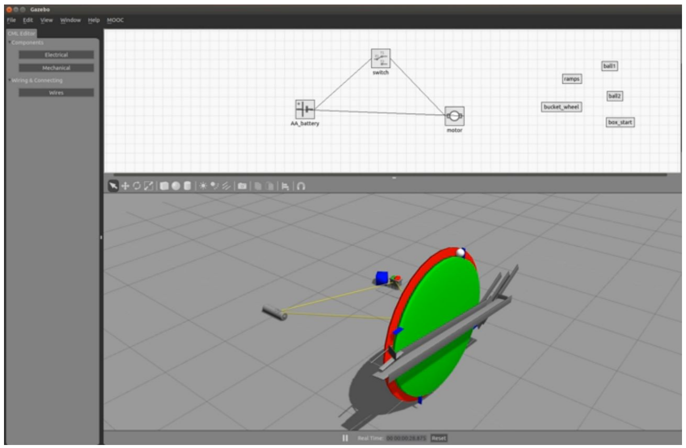
MENTOR2: Gazebo Design Kit (GDK) (1 of 3)



Improved GUI with model editing capabilities

Joint creation
Alignment tools

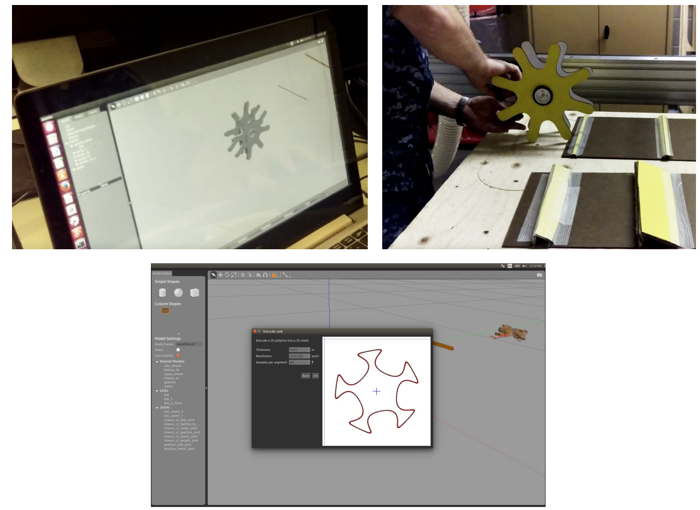
Copy/paste
Undo



2D schematic view to illustrate connections between components

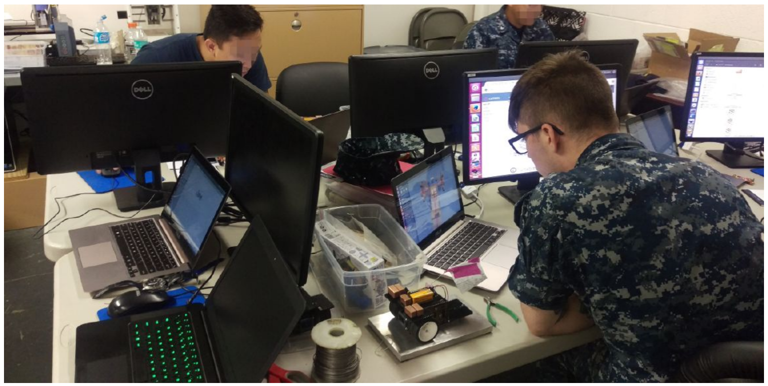
MENTOR2: Gazebo Design Kit (GDK) (2 of 3)

Provides real time feedback through an online learning companion



Supports importing laser-cutter files

MENTOR2: GDK User Evaluation (3 of 3)



Real world, hands-on testing with non-traditional users

Three 12-hr workshops with US Navy

Design Challenges

- Build a vehicle to climb incline
- Modify vehicle to drag weight
- Design & laser cut new wheels to traverse bumps

Deployable Buildfarm

Custom ROS buildfarm

BOSCH

Bosch Buildfarm: Overview

The ROS buildfarm:

- Automatically builds .deb files from your packages
- Handles continuous integration (unit tests)
- Automatically creates documentation (Doxygen, Sphinx, Epydoc, etc.)

A deployable buildfarm enables:

- Code hosting on private servers (i.e. you can't use public GitHub)
- Distribution of proprietary ROS packages (only) to customers
- Maintenance of specific package versions (e.g. for stability)

ROS BOSCH

Bosch Buildfarm & OSRF

Project Description

Support deployment of the ROS buildfarm for internal use at Bosch.

This gives Bosch more control over their code, and supports proprietary packages.

The resulting product is an open source implementation available to anyone.

Timeframe

Oct 2014 – Jan 2015

Funding Source

BOSCH

Bosch Buildfarm: Technical details

Technologies Used

Jenkins

The continuous integration application coordinating the processes.

Docker

The container technology to provide build isolation.

Reprepro

The apt repository management tool to manage the resulting packages.

ROS on ARM

ROS on ARM: Overview

ARM is a family of CPU known for low power consumption.

This makes ARM processors:

- The default choice for mobile applications
- Valuable for any battery-powered system, such as robots

Qualcomm is a big player in the mobile space because of their Snapdragon ARM-based processors. They are supporting the development & testing of ROS on ARM-based processors for use in robotics.

ROS on ARM & OSRF

Project Description

Make ROS available for use on ARM-based platforms.

This makes it possible to run ROS on low-power, single-board computers as well as smartphones and tablets.

Timeframe

Oct 2014 – Jan 2015

Funding Source

Qualcomm

ROS on ARM: Technical details

Technologies Used

QEMU

Allows emulation of ARM environments on x86-based build machines.

Docker

Provides build isolation for packages to support different architectures in conjunction with qemu.

ARM-H

ARM-H & OSRF

Project Description

OSRF designed the electronics, firmware, and software for the (relatively) low-cost hands designed by Sandia National Laboratories.

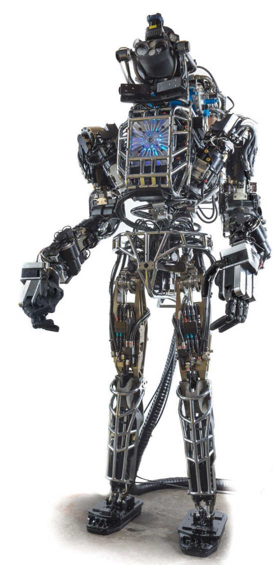
The hands were one of the options that teams could use on the Atlas robots during the DRC Trials.

Timeframe

2010 – 2013

Funding Source

DARPA

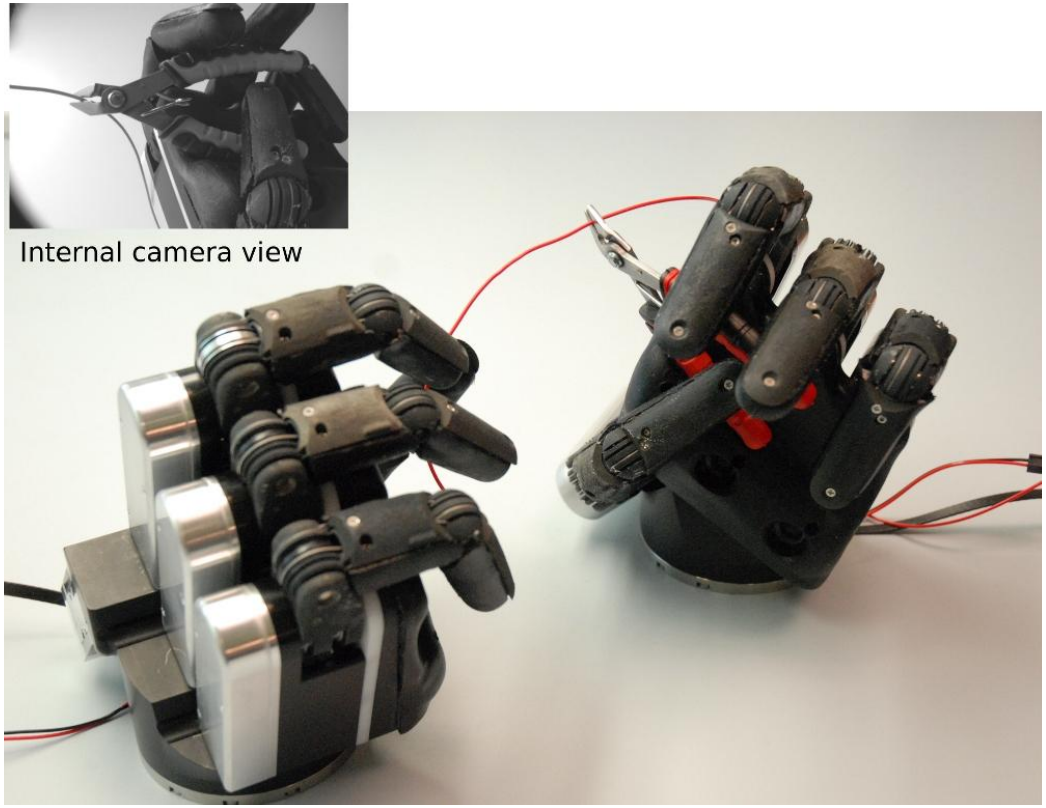


ARM-H: Overview (1 of 2)

Each finger is a self-contained "micro-arm".

In fact, fingers could fall off while the hand is operating, without affecting the other fingers!

ARM-H: Overview (2 of 2)



The hands have cameras, inertial sensors, and tactile sensors.

M3-Actuators

M3-Actuators & OSRF

Project Description

The program goal was to increase the walking efficiency of full-size humanoid robots by an order of magnitude.

We collaborated with Sandia National Laboratories and IHMC to build two humanoids.

Timeframe

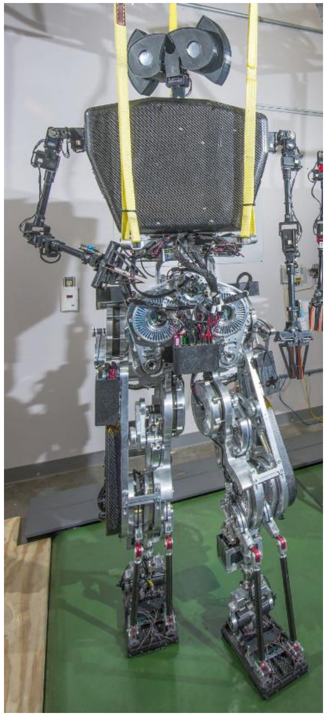
2013 – 2015

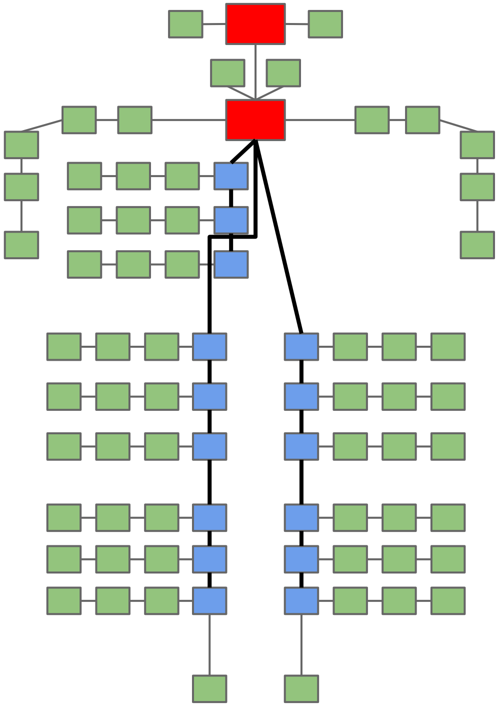
Funding Source

DARPA

M3-Actuators: Overview (1 of 2)

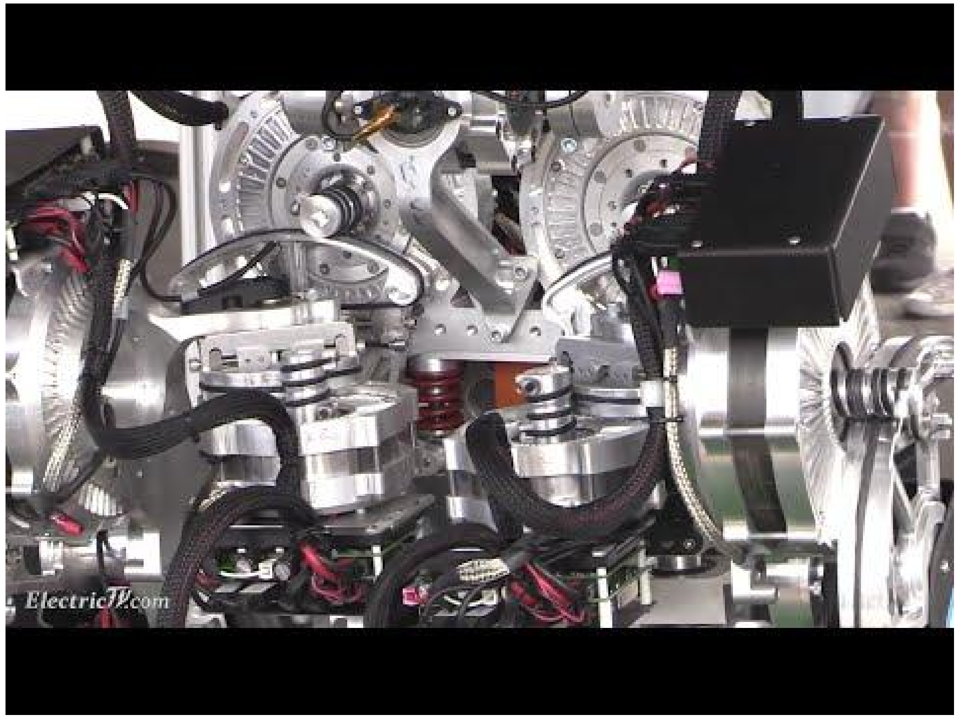
Each robot is a large network of custom electronics.
This robot has over 60 processors and 15 FPGA's !





Red: x86
Green: MCU's
Blue: FPGA's

M3-Actuators: Overview (2 of 2)



https://www.youtube.com/watch?v=Kvk3y6OdXA

At the DRC Finals Expo, the final robot walked for 4 hours, covering 2.8 km, on a single battery charge! The average power draw was only 450 watts.

ROS 2

ROS 2: Overview (1 of 2)

ROS 2 will add support for scenarios where ROS 1 falls short. This includes:

- Communication over lossy networks (e.g. wireless)
    - multi-robot systems, drones, etc.

- Support for real-time

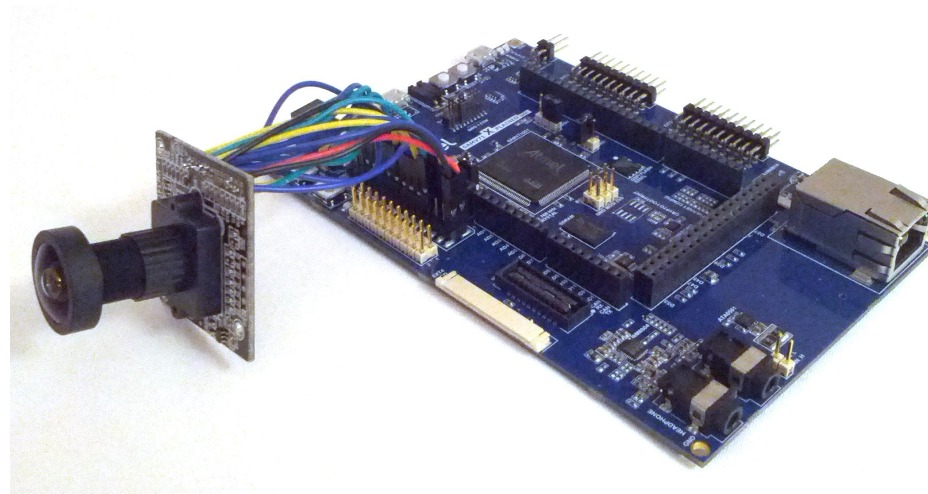
ROS 2: Overview (2 of 2)

ROS 2 will add support for scenarios where ROS 1 falls short. This includes:

- Support for Windows, Linux and OSX
- Use of DDS to bridge the gap between prototyping and product shipment
- Support for small embedded systems

ROS 2: Embedded-friendly

The network protocols behind ROS 2 can be implemented on small processors, too!



This demo shows a single-chip ARM microcontroller sending camera images directly to other ROS programs.

ROS 2 & OSRF

Project Description

Research and evaluate directions for ROS 2.

Design and specify the ROS 2 system architecture.

Implement ROS 2.

Timeframe

Jan 2014 – Dec 2016

Funding Source

NASA NSF

ROS 2: Technical details

Technologies Used

C++11

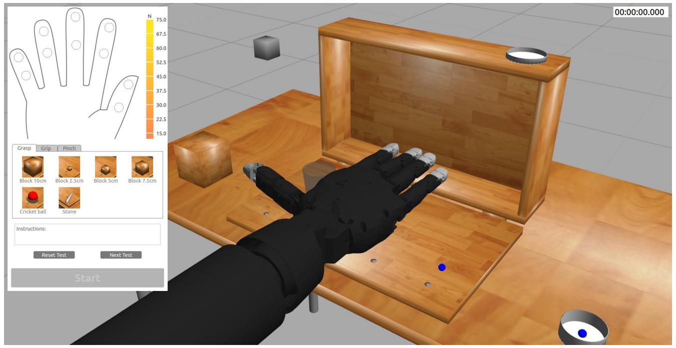
- std smart pointer (instead of boost)
- std::chrono
- Easier cross-platform support
- Cleaner code

Python 3

- New packages like asyncio
- Python 2 is in bug-fix-only mode

HAPTIX

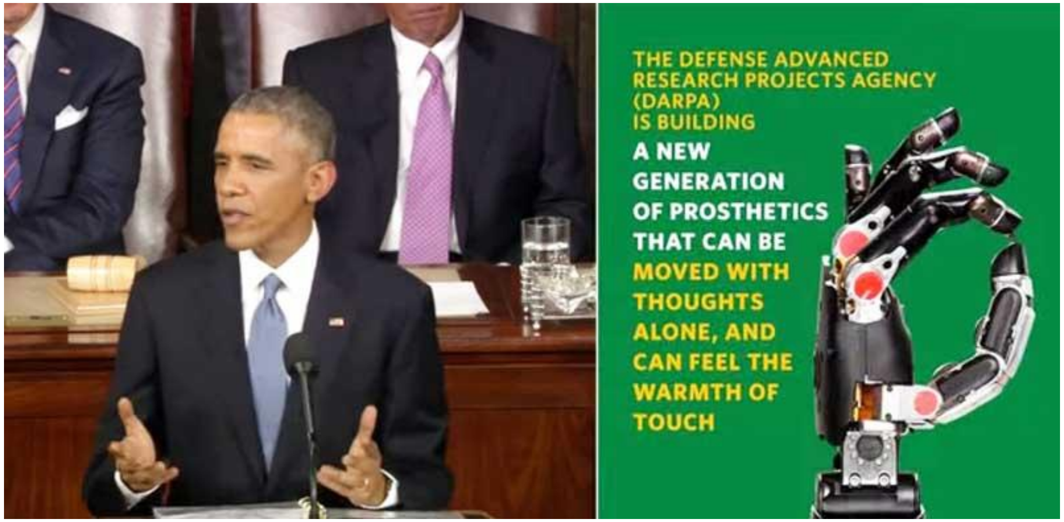
Hand Proprioception & Touch Interfaces



Open Source Robotics Foundation

HAPTIX: Overview

The goal of the program is to develop neural interfaces that allow transradial amputees to control and sense through advanced robotic prosthetic limbs.



President Obama described HAPTIX at the State of the Union Address

HAPTIX & OSRF

Project Description

OSRF is providing realistic prosthetic simulation and virtual test environments for interface testing and controls software development.

Timeframe

2014 – 2017

Funding Source

DARPA

HAPTIX: Hardware Overview

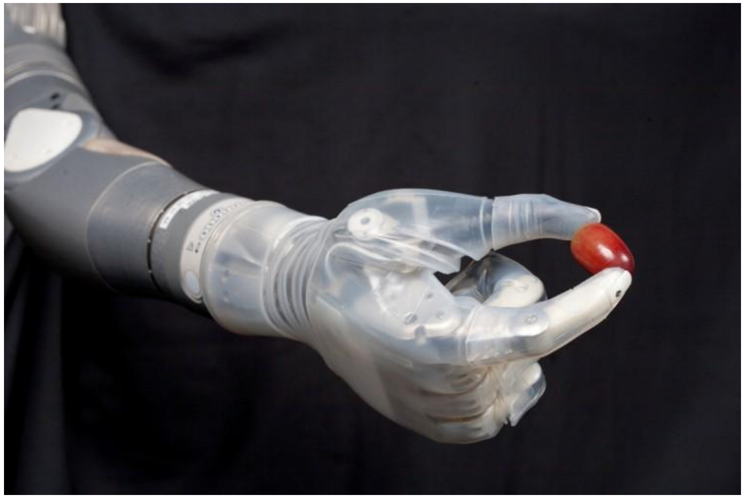
DEKA "Luke Hand"

- Physical hardware is being developed in parallel at DEKA.

- Complex transmission mechanisms to mimic human hand control (14 DOF hand driven by 5 electric motors).

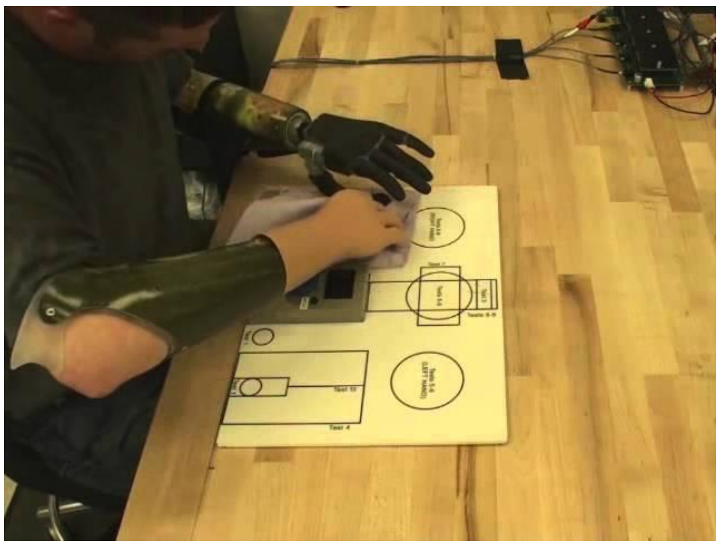
- Electric clutch at every actuator to save power.

- CAN bus interface.



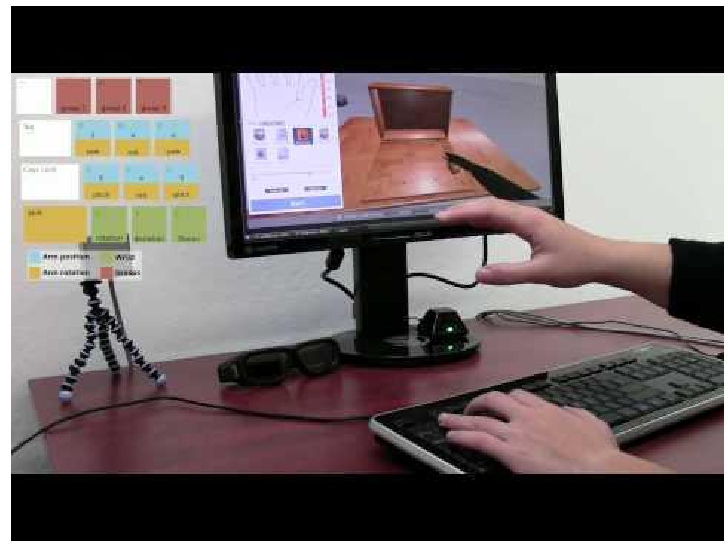
	DOFs	Actuators
wrist	2	1
thumb	2	2
index	2	1
middle/ ring/pinky	8	1

HAPTIX: Videos



https://www.youtube.com/watch?v=WIglB2zeyOE

Example test environment video (SHAP)



https://www.youtube.com/watch?v=vm9v5Z-eOxg

Gazebo simulation demo