Introduction to Autonomous Robotics

Michael McCool
Principal Engineer, Intel Corporation

Richard Pekkala
Design Engineer, Intel Corporation

Agenda

- IoT and Robotics
- Intel® RealSense™ Robotic Development Kit
- Robot HW and SW Architecture
- ROS* Concepts and Navigation
- *Practicum*
- Summary and Next Steps

Agenda

- **IoT and Robotics**
- Intel® RealSense™ Robotic Development Kit
- Robot HW and SW Architecture
- ROS* Concepts and Navigation
- *Practicum*
- Summary and Next Steps

IDF16
INTEL DEVELOPER FORUM

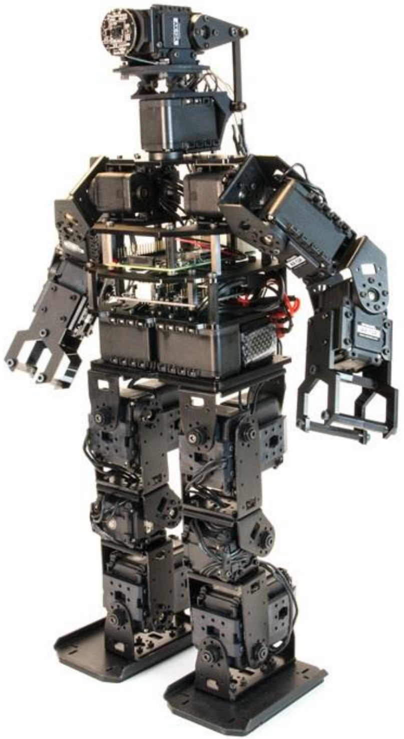
Relationship Between the IoT and Robotics

Robots are…

- Autonomous systems with the capability to interact with their environments
  - Not just passive sensors
- *If* connected to the internet, can also participate in the IoT ecosystem
  - Share experiences via the network
  - Use sensor data from other IoT devices
  - Offload work to remote systems
  - Access large amounts of data
  - Coordinate motion across multiple robots/devices

*Next-generation **connected** robots are explicitly leveraging IoT technologies and middleware*

*But they also need to be **autonomous***

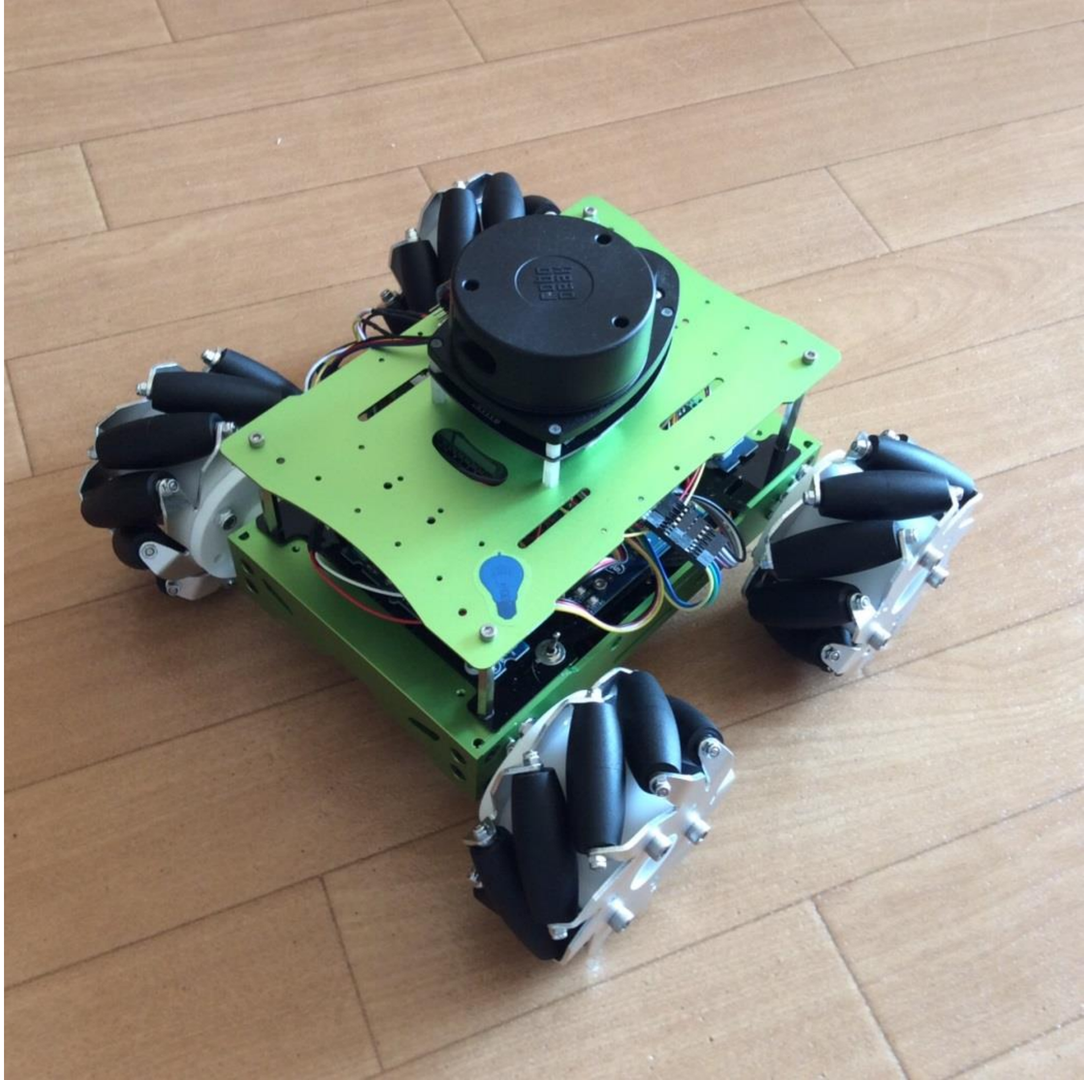


IDF16 INTEL DEVELOPER FORUM

Mobile Robotics

Mobile robots need to:

- Sense a potentially dynamic environment
- Determine current location
- Navigate from one location to another
- Interact with humans as needed (avoid, follow, respond to commands, etc.)



Social Robotics

Robots that interact with people need specialized skills:

- Speech recognition
- Face recognition
- Dialogue management
- Context awareness
- Emotional state recognition
- Social relationship understanding

Agenda

- IoT and Robotics
- **Intel® RealSense™ Robotic Development Kit**
- Robot HW and SW Architecture
- ROS* Concepts and Navigation
- *Practicum*
- Summary and Next Steps

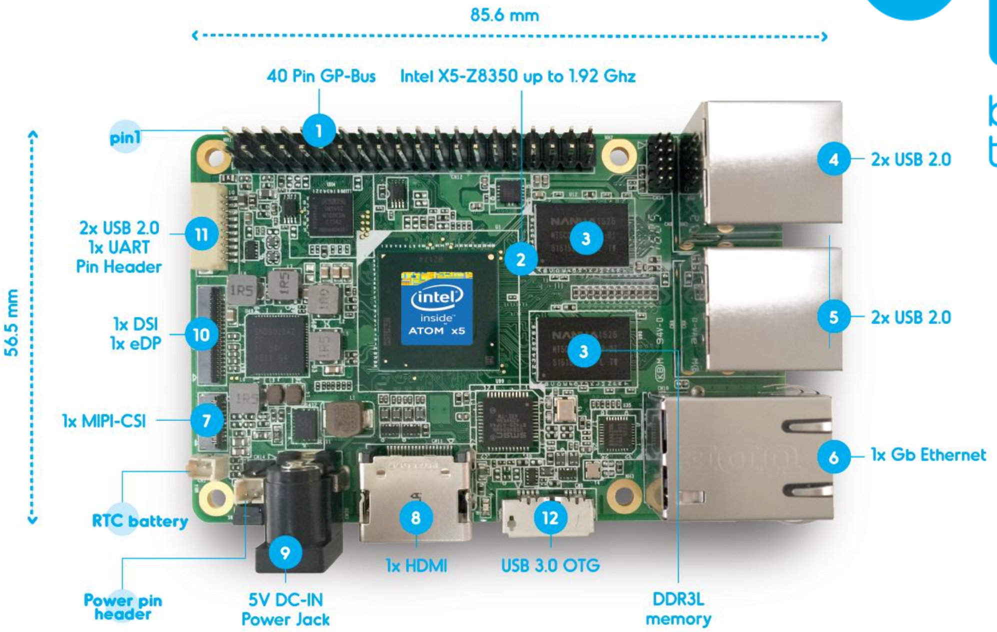
IDF16
INTEL DEVELOPER FORUM

Intel® RealSense™ Robotic Development Kit

Aaeon* UP* Board

Powerful new development board based on quad-core Intel® Atom™ x5Z8350

- Clock scales from 490Mhz to 1.9Ghz
- 1, 2 or 4 GB of DDR3L RAM
- 16, 32 or 64 GB of eMMC storage
- 4x USB2.0 ports, +2x USB2 on header
- 1x USB3.0 OTG port
- HDMI*
- 40-pin connector with I2C, SPI, UART, I2S, PWM, and ADC; driven by Altera® MAX™ V CPLD
- MIPI-DSI, MIPI-CSI, embedded DisplayPort* (eDP)
- Wired Gigabit Ethernet

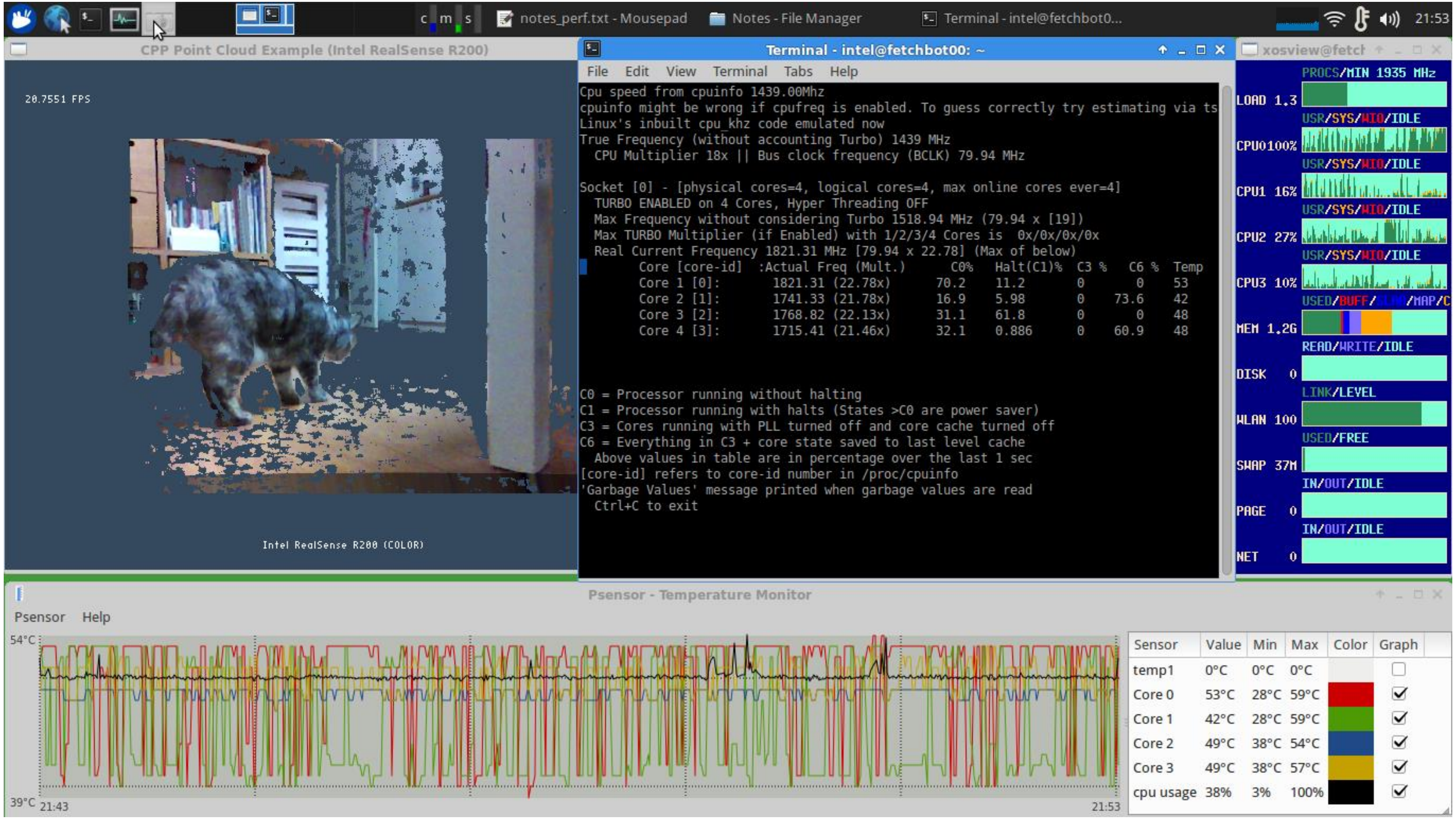


- OS-64bit: Linux*, Android*, Windows* (IoT and 10)
- Intel® Gen 8 graphics supporting OpenCL*
- ROS* and Intel® RealSense™ supported

Intel® RealSense™ R200 Camera

- Provides RGBD (color and depth) data at a high frame rate (30Hz to 90Hz) and resolution
- Requires USB3.0 interface
- Open-source Linux* drivers (librealsense) and ROS* wrappers now available

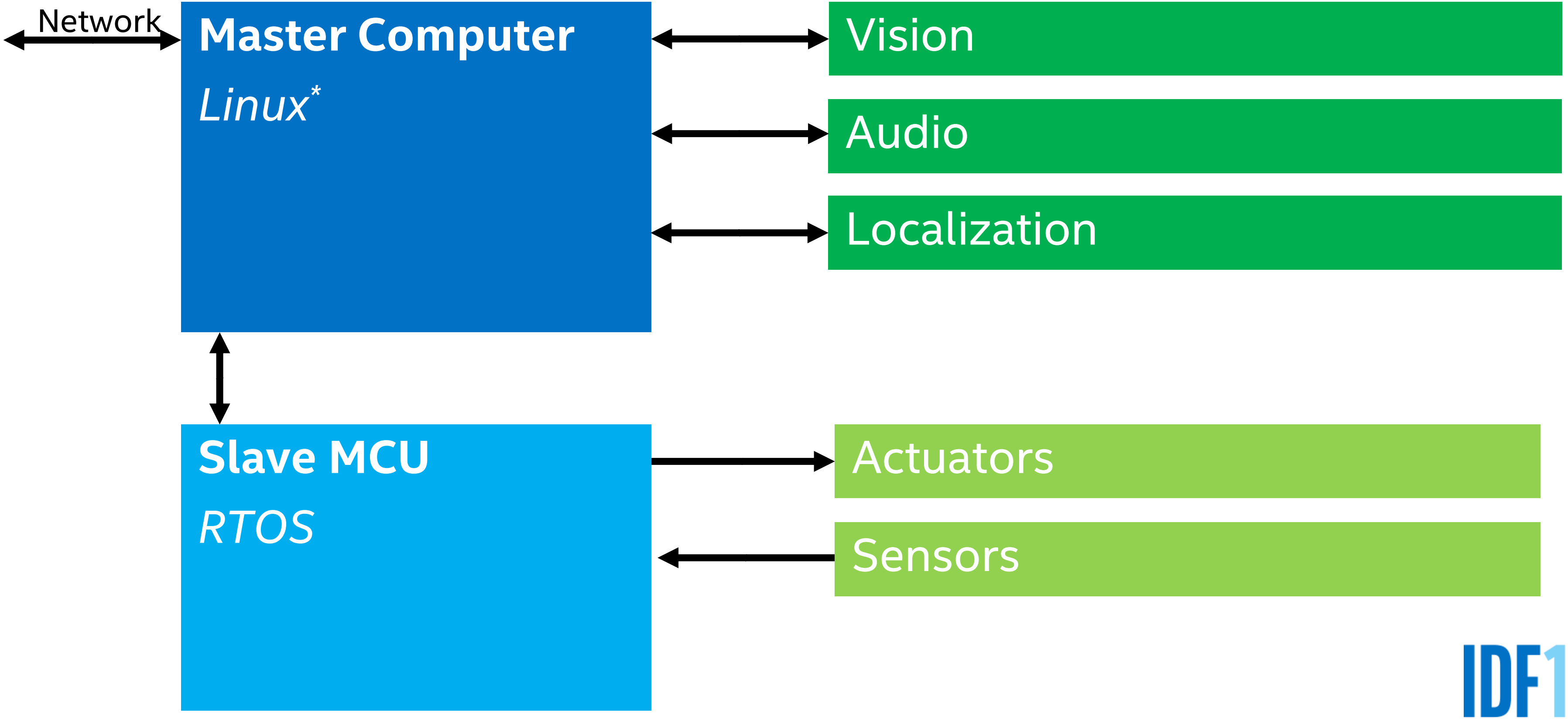
An Example...


notes_perf.txt - Mousepad
Notes - File Manager
Terminal - intel@fetchbot0...
21:53
CPP Point Cloud Example (Intel RealSense R200)
20.7551 FPS
Intel RealSense R200 (COLOR)
Terminal - intel@fetchbot00: ~
File Edit View Terminal Tabs Help
Cpu speed from cpuinfo 1439.00Mhz
cpuinfo might be wrong if cpufreq is enabled. To guess correctly try estimating via ts
Linux's inbuilt cpu_khz code emulated now
True Frequency (without accounting Turbo) 1439 MHz
CPU Multiplier 18x || Bus clock frequency (BCLK) 79.94 MHz
Socket [0] - [physical cores=4, logical cores=4, max online cores ever=4]
TURBO ENABLED on 4 Cores, Hyper Threading OFF
Max Frequency without considering Turbo 1518.94 MHz (79.94 x [19])
Max TURBO Multiplier (if Enabled) with 1/2/3/4 Cores is 0x/0x/0x/0x
Real Current Frequency 1821.31 MHz [79.94 x 22.78] (Max of below)
Core [core-id] :Actual Freq (Mult.) C0% Halt(C1)% C3 % C6 % Temp
Core 1 [0]: 1821.31 (22.78x) 70.2 11.2 0 0 53
Core 2 [1]: 1741.33 (21.78x) 16.9 5.98 0 73.6 42
Core 3 [2]: 1768.82 (22.13x) 31.1 61.8 0 0 48
Core 4 [3]: 1715.41 (21.46x) 32.1 0.886 0 60.9 48
C0 = Processor running without halting
C1 = Processor running with halts (States >C0 are power saver)
C3 = Cores running with PLL turned off and core cache turned off
C6 = Everything in C3 + core state saved to last level cache
Above values in table are in percentage over the last 1 sec
[core-id] refers to core-id number in /proc/cpuinfo
'Garbage Values' message printed when garbage values are read
Ctrl+C to exit
xosview@fetch
PROCS/MIN 1935 MHz
LOAD 1.3
USR/SYS/WIO/IDLE
CPU0 100%
CPU1 16%
CPU2 27%
CPU3 10%
MEM 1.2G
READ/WRITE/IDLE
DISK 0
LINK/LEVEL
WLAN 100
USED/FREE
SWAP 37M
IN/OUT/IDLE
PAGE 0
NET 0
Psensor - Temperature Monitor
Psensor Help
54°C
39°C
21:43
21:53
Sensor Value Min Max Color Graph
temp1 0°C 0°C 0°C
Core 0 53°C 28°C 59°C
Core 1 42°C 28°C 59°C
Core 2 49°C 38°C 54°C
Core 3 49°C 38°C 57°C
cpu usage 38% 3% 100%

Agenda

- IoT and Robotics
- Intel® RealSense™ Robotic Development Kit
- **Robot HW and SW Architecture**
- ROS* Concepts and Navigation
- *Practicum*
- Summary and Next Steps

Generic Robot Architecture



Example 2WD Robot

Computer: *UP* Board*, using an Intel® Atom™ x5 processor (Quad-core), with 4GB RAM, 32GB storage, wired Gb network, 4x USB2.0 and 1x OTG USB3.0 ports

Camera: Intel® RealSense™ R200 3D Depth Camera

Actuation: Two Dynamixel MX-12W smart servos with magnetic encoders

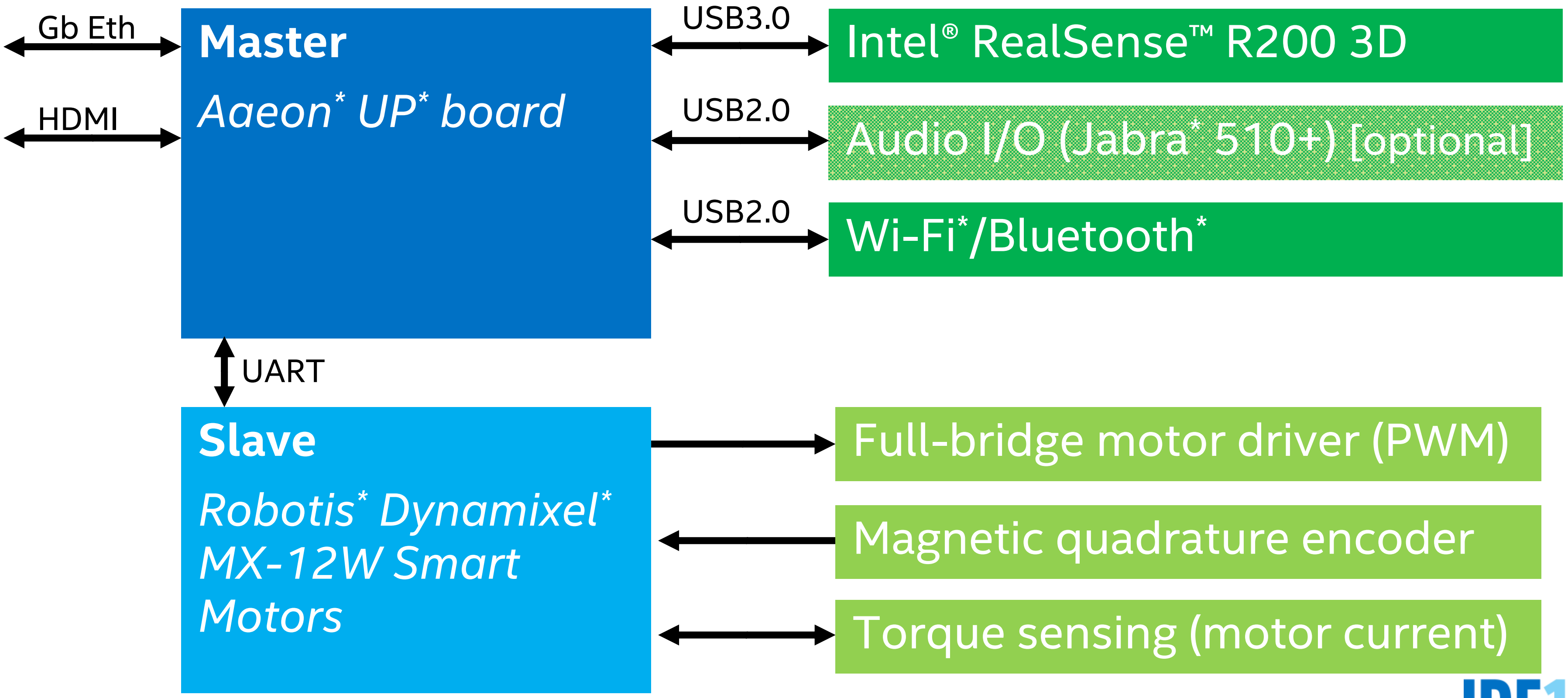
Software: Xubuntu 14.04 and ROS* Indigo

Frame: Laser-cut acrylic, Polulo sphere casters, o-ring tires and belt transmission

Other: DF Robot power regulator

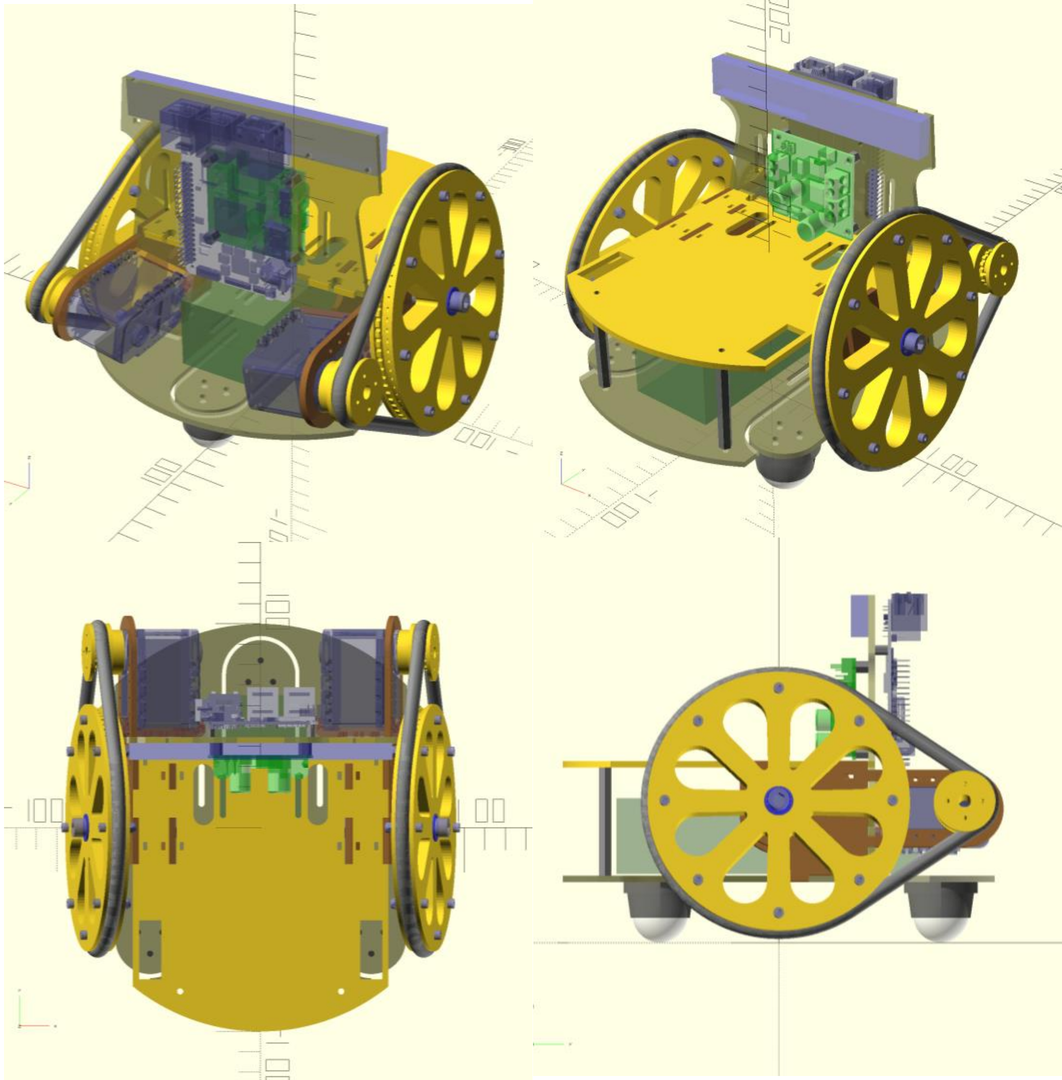
Extras: Jabra* 510+ USB speakerphone

Example 2WD Robot Architecture



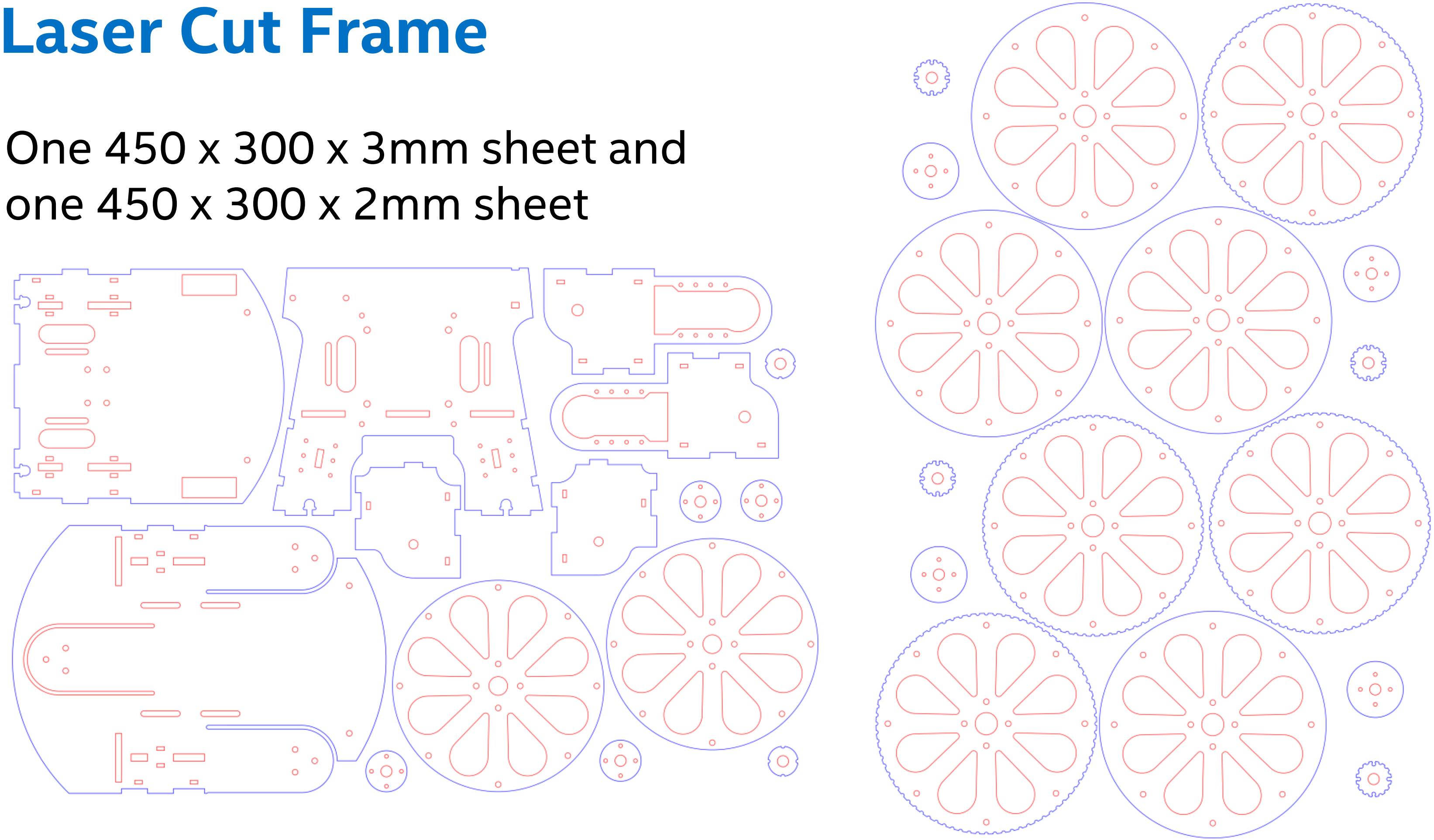
Development Process

- Software stack based on modification and simplification of the OSRF* Turtlebot* stack
- Custom motor driver written based directly on Dynamixel* Linux* SDK
- Frame parametrically modelled using OpenSCAD* and then converted to lasercut files using Inkscape*



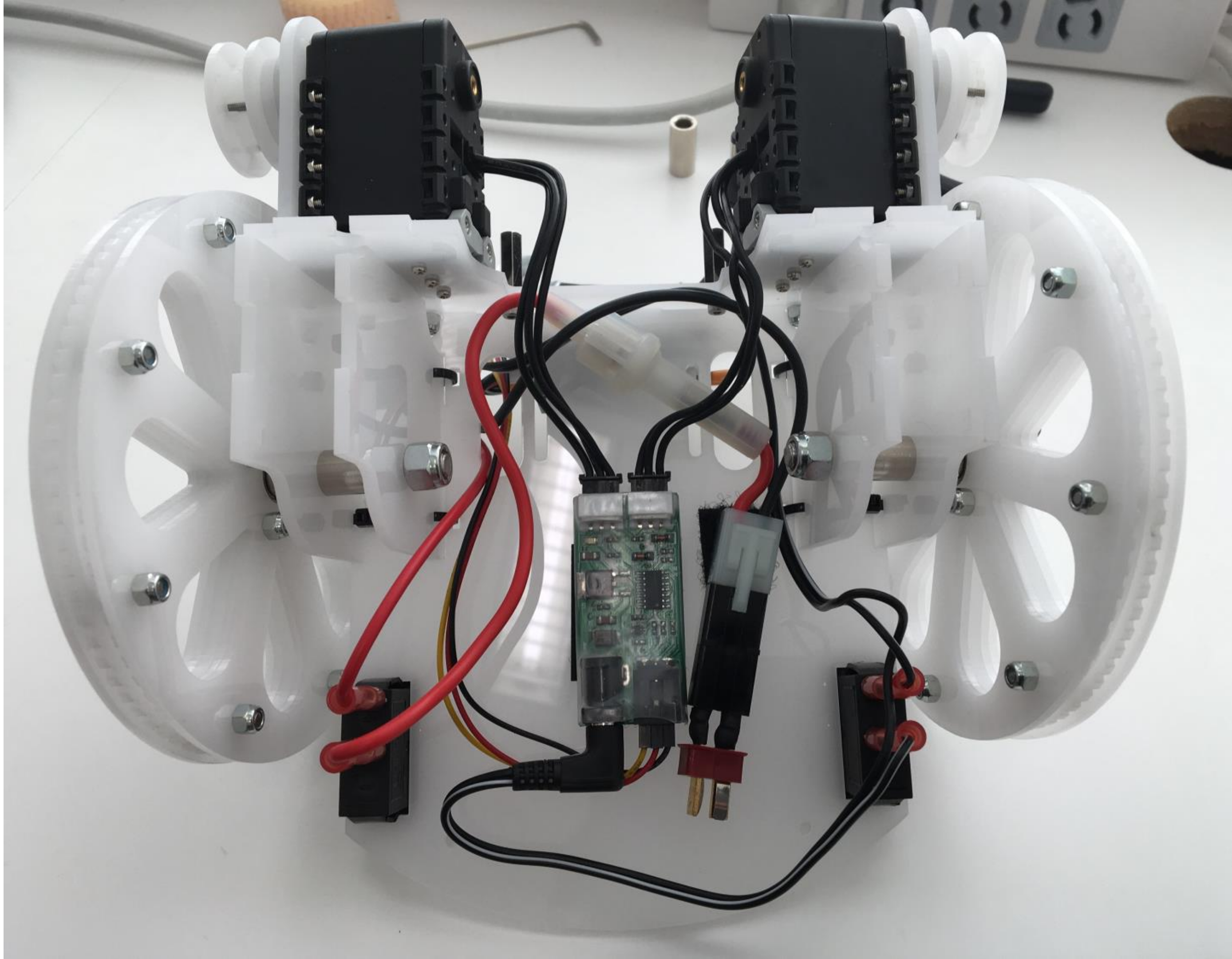
Laser Cut Frame

One 450 x 300 x 3mm sheet and
one 450 x 300 x 2mm sheet



Assembly…

- Working on detailed online instructions…



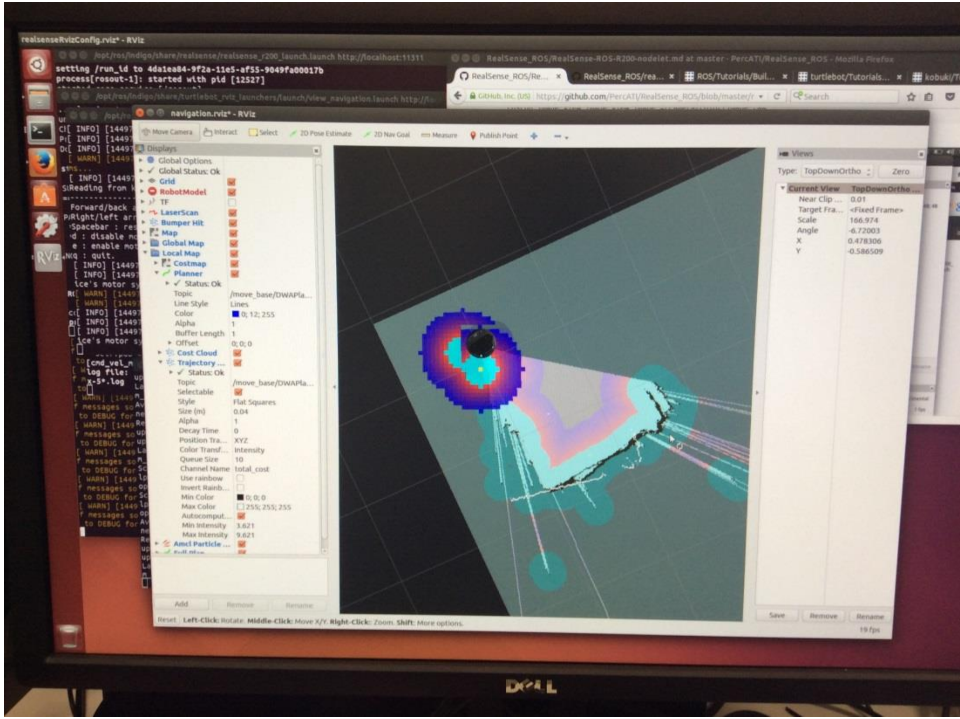
Agenda

- IoT and Robotics
- Intel® RealSense™ Robotic Development Kit
- Robot HW and SW Architecture
- **ROS* Concepts and Navigation**
- *Practicum*
- Summary and Next Steps

ROS*: Robot Operating System

ROS

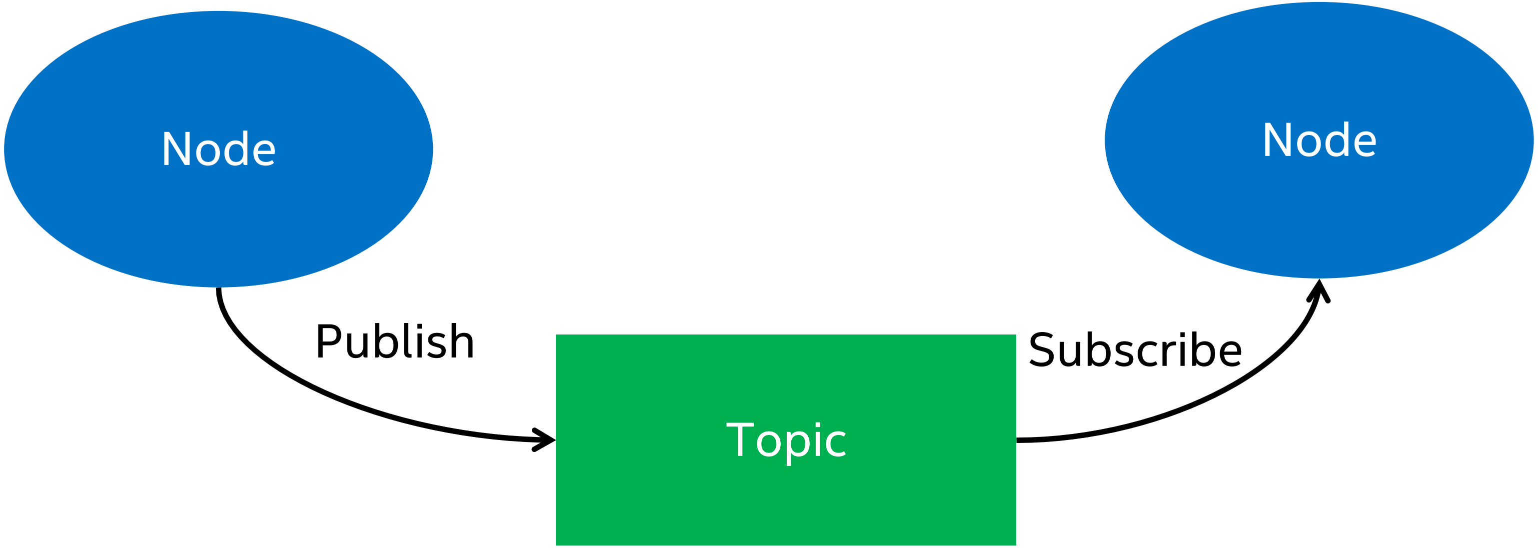
- Supports distributed system based on a graph of communicating nodes
- Large community with many existing nodes for sensors, planning, navigation, etc.
- Various visualization tools and simulators are also provided



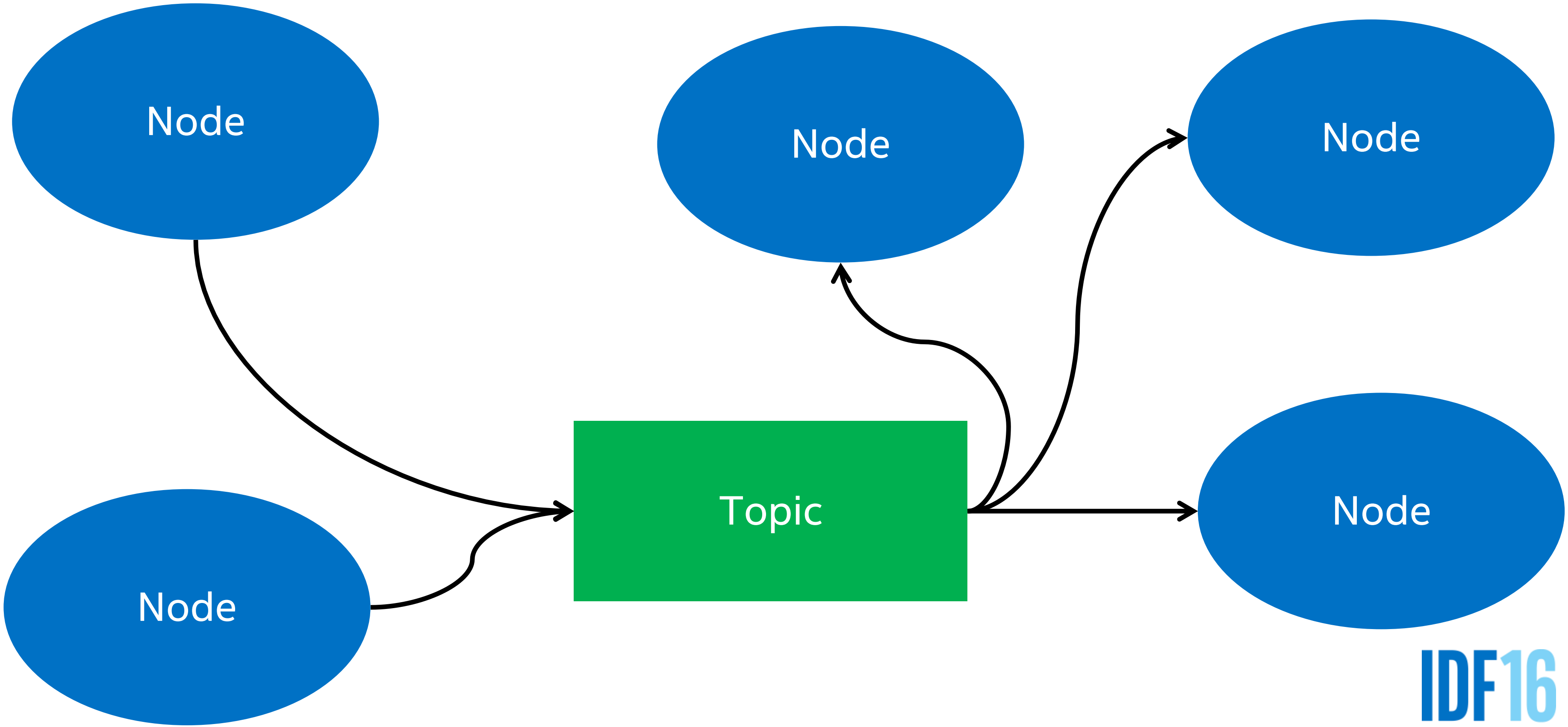
IDF16
INTEL DEVELOPER FORUM

Basic ROS* Concepts: Messages and Topics

- Publish/subscribe data on uniquely named topics
- Topics can have multiple publishers and subscribers
- Messages are typed, can carry multiple elements

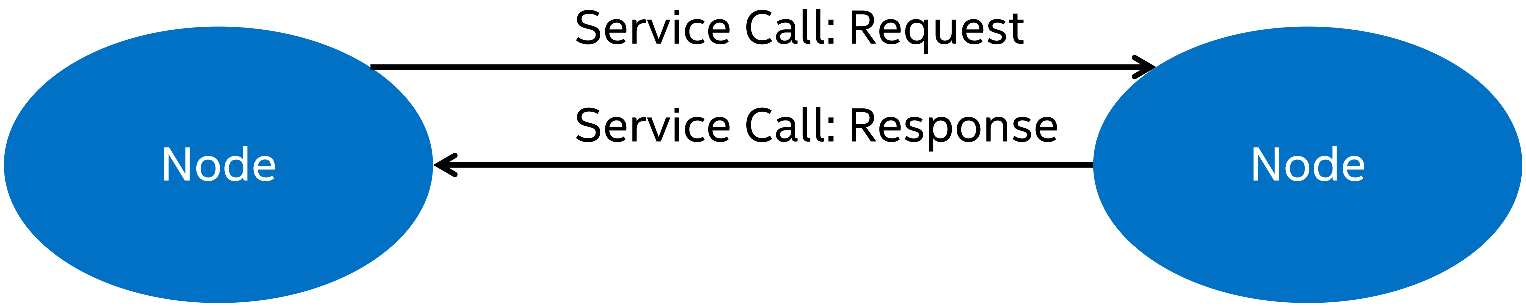


Basic ROS* Concepts: Messages and Topics



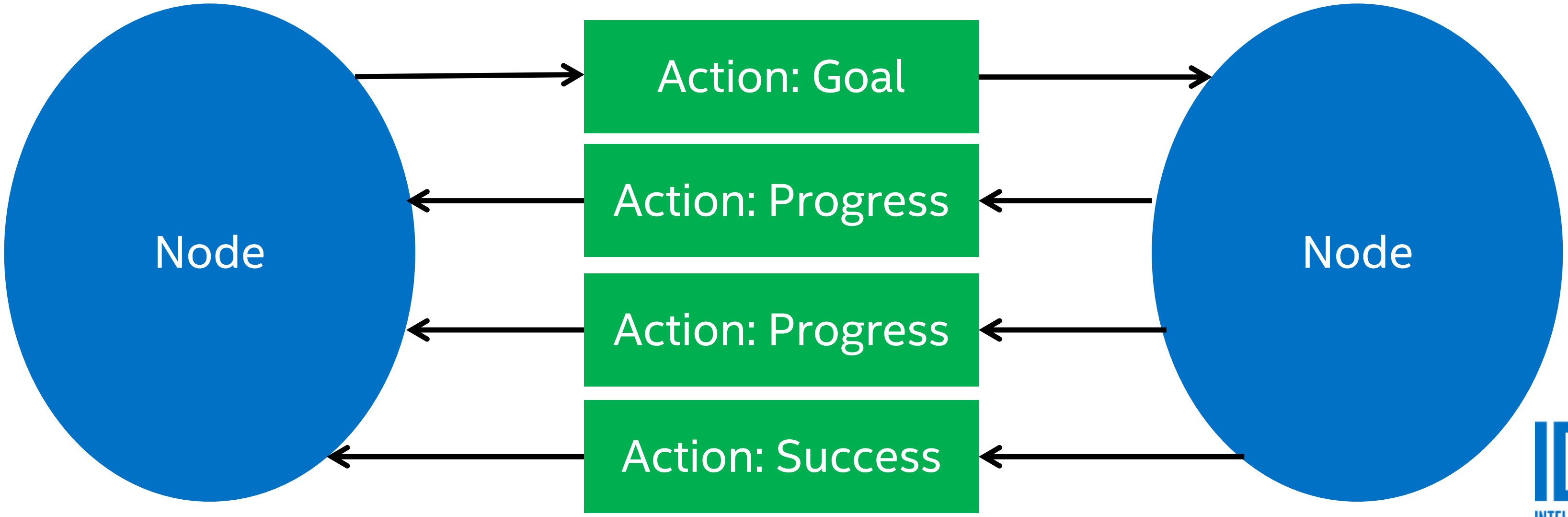
Basic ROS* Concepts: Service Calls

- Synchronous remote procedure call
- Caller blocks until response received
- Of limited use: synchronization, get/set



ROS* Concepts: Actions

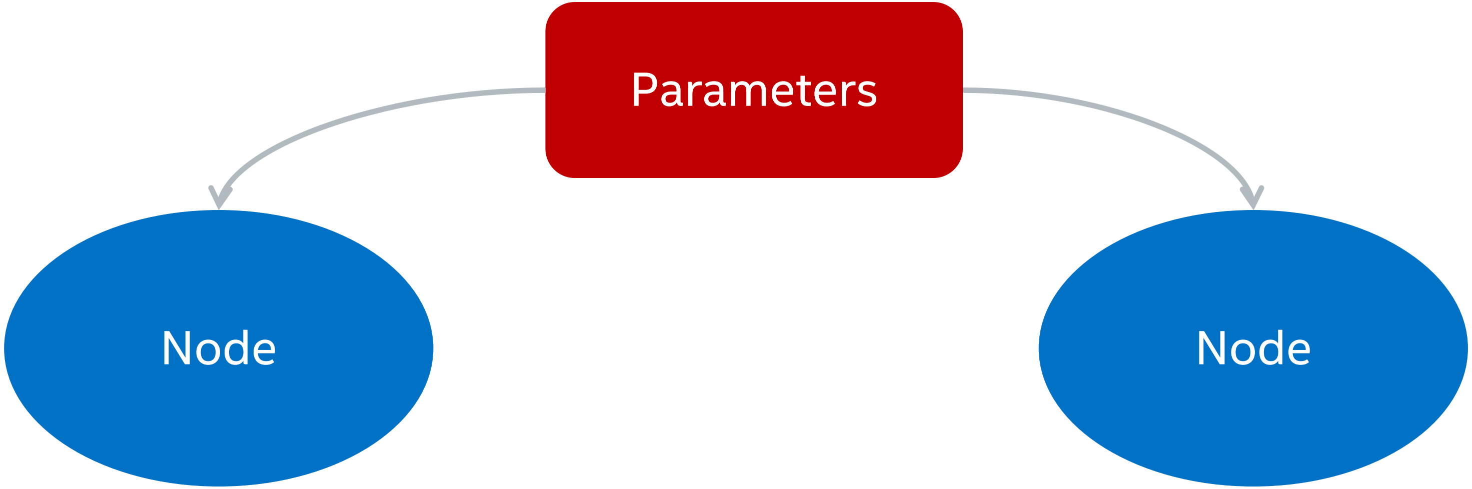
- Asynchronous goal-directed behavior control based on standard set of topics
- Set goals, initiate actions, monitor progress, cancel if necessary, be informed of success or failure



IDF16
INTEL DEVELOPER FORUM

Basic ROS* Concepts: Parameters

- Parameter server provides typed constant data at startup
- Parameters can be specified in various ways: command line, parameter files, launch file parameters, etc.
- Newer version of ROS provide dynamic parameter update



IDF16 INTEL DEVELOPER FORUM

Other ROS Concepts

Packages:

- Collections of files used to implement or specify a service or node in ROS*, built together using catkin (typically)

URDF (Universal Robot Description):

- XML files describing joints and transformations between joints in a 3D model of robot

Launch Files:

- XML files describing a set of nodes and parameters for a ROS graph

YAML (Yet Another Markup Language):

- Used for parameter specification on the command line and in files

ROS* Tools

Rviz:

- Visualize various forms of dynamic 3D data in context: transforms, maps, point clouds, images

Gazebo:

- Robot simulation, including collisions, inertia, perceptual errors, etc.

Rqt:

- Visualize graphs of nodes and topics

Command-line tools:

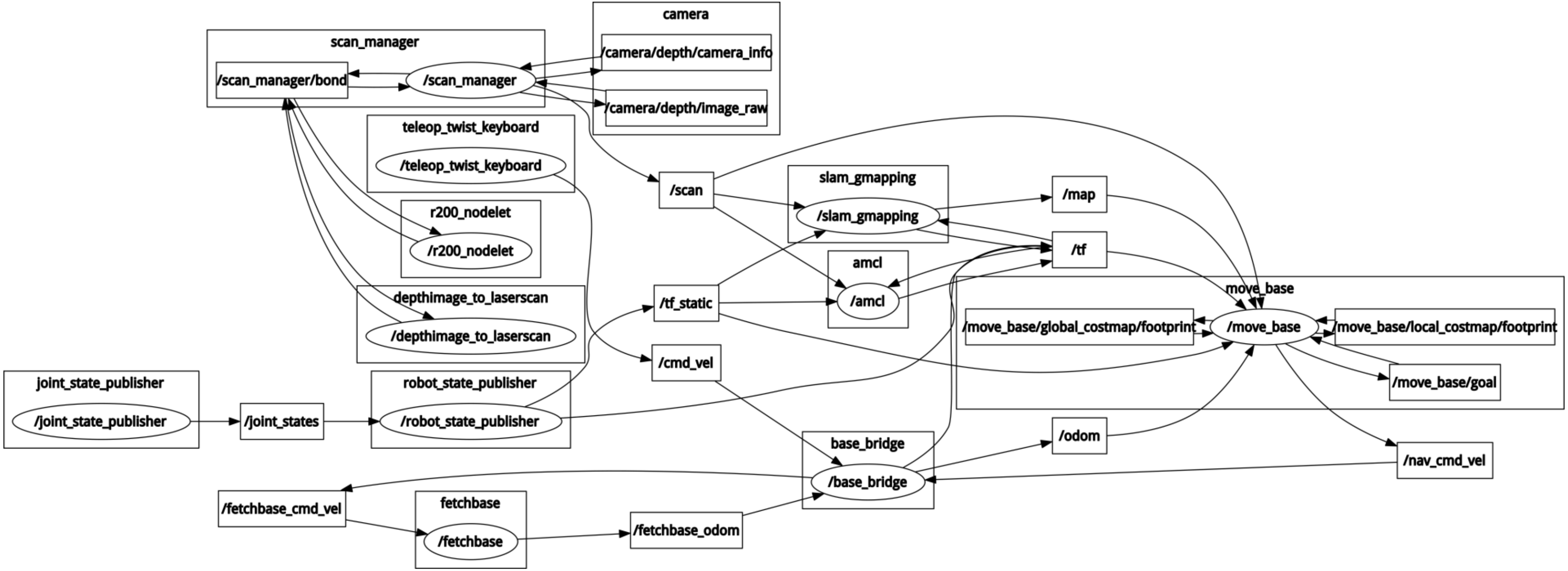
- Listen to and publish on topics, make service calls, initiate actions
- Filter and monitor error messages

Catkin:

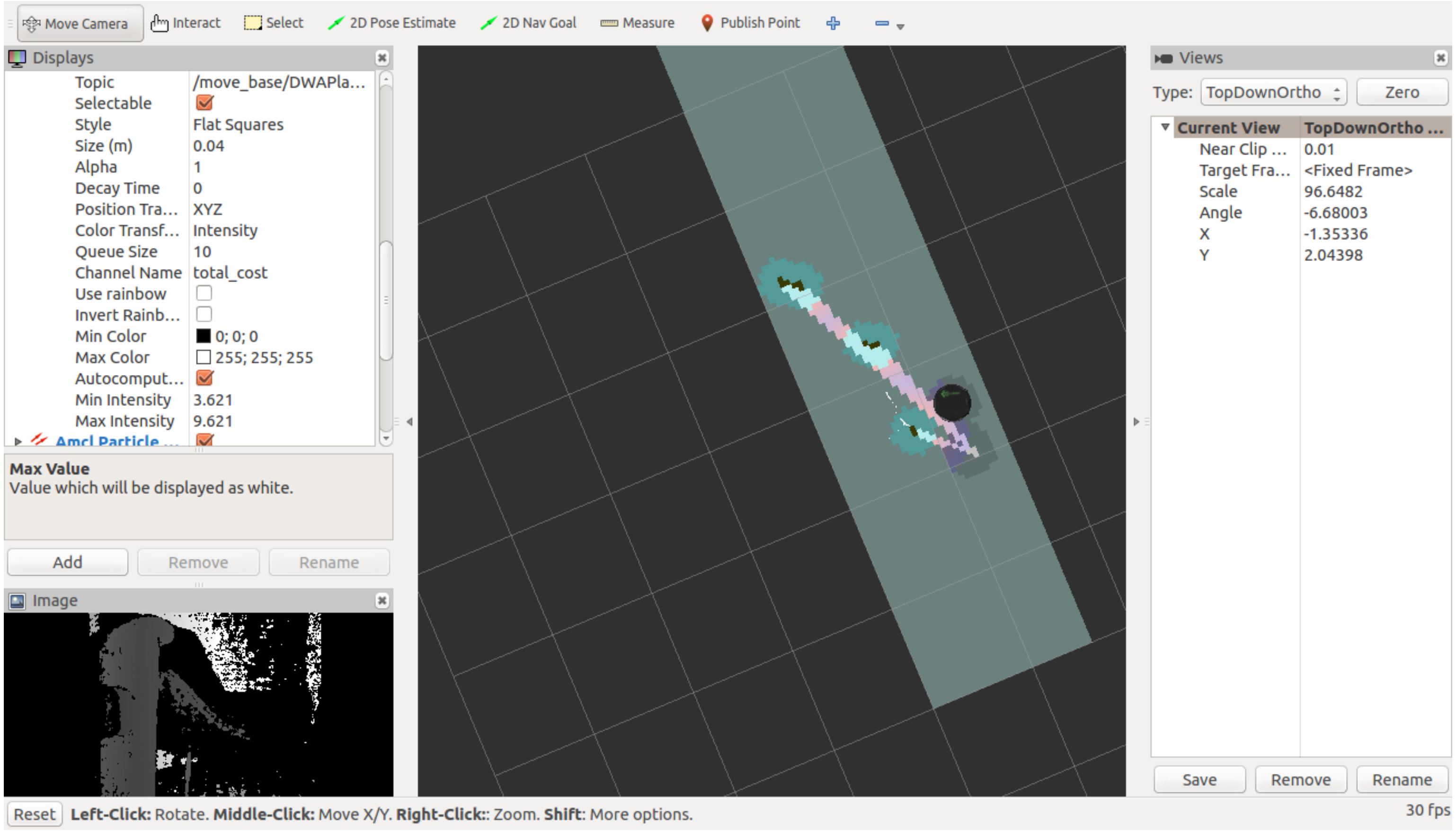
- Build system and package management

IDF16 INTEL DEVELOPER FORUM

Autonomous Navigation ROS* Node Graph



Rviz: Robot Visualization



Agenda

- IoT and Robotics
- Intel® RealSense™ Robotic Development Kit
- Robot HW and SW Architecture
- ROS* Concepts and Navigation
- ***Practicum***
- Summary and Next Steps

IDF16
INTEL DEVELOPER FORUM

Practicum

- Refer to **Instructions.pdf** on your desktop
- You will
  - Log into robot
  - Start up ROS* navigation stack
  - Test robot motor control using teleoperation
  - Explore ROS command line and graphical debugging tools
  - Look at some example launch and parameter files
  - Look at some example C++ code for a motor driver
  - Launch Rviz on the host to visualize robot telemetry
  - Build a map using scan data
  - Allow the robot to navigate autonomously (if Wi-Fi* is willing)

Agenda

- IoT and Robotics
- Intel® RealSense™ Robotic Development Kit
- Robot HW and SW Architecture
- ROS* Concepts and Navigation
- *Practicum*
- **Summary and Next Steps**

IDF16
INTEL DEVELOPER FORUM

Summary and Next Steps

- Autonomous robotics is an emerging *product* area
- Technology for mobile robotics is now relatively mature
- Robotics involves all aspects of computer science and engineering
- Overlaps with but is distinct from IoT
- ROS* is a useful framework for robot development

NEXT STEPS:

→Install ROS and try it with a simulator

→Try ROS on a robot of your own

→Develop new robotic applications

IDF16 INTEL DEVELOPER FORUM

Additional Sources of Information

- A PDF of this presentation is available from our Technical Session Catalog: www.intel.com/idfsessionsSF. This URL is also printed on the top of Session Agenda Pages in the Pocket Guide.

- OSRF: http://www.osrfoundation.org/
- ROS*: http://ros.org and http://wiki.ros.org
- Recommended books:
  1. *Programming Robots with ROS: A Practical Introduction to the Robot Operating System*, by Morgan Quigley and Brian Gerkey. O'Reilly, 2015.
  2. *Learning ROS for Robotics Programming*, 2nd Edition, by Enrique Fernandez and Luis Sanchez Crespo, PACKT, 2015
  3. *Mastering ROS for Robotics Programming*, by Lentin Joseph, PACKT, 2015

Technical Sessions in this Track

Wednesday, April 13, 2016

NDSTI01 — Intel® RealSense™ Technology: Adding Human-like Sensing to Devices

NDSTS03 — Intel® Robotics Overview

NDSTS05 — Getting Started with the Intel® RealSense™ Robotic Development Kit

Thursday, April 14, 2016

MAKE010 — Introduction to Perceptual Computing Using Intel® RealSense™ & the Intel® Joule™ Compute Module
Level 6 Room Auditorium, 1:15 PM – 2:15 PM

IDF16 INTEL DEVELOPER FORUM

Technical Sessions in Software Track

Tuesday, August 16, 2016

11:00 AM – 12:00 PM *SOFTS01* — Accelerating Machine Learning on Apache Spark* ***Level 2 Room 2006***
1:15 PM – 2:15 PM *SOFTS02* — ChromeOS* and coreboot* on Intel® Architecture – An Engineering Primer for Developers, Partners, OEMs and ODMs ***Level 2 Room 2006***
2:30 PM – 3:30 PM *ANATS05* — How to Parallelize Neural Networks (xNNs) for Intel® Xeon Phi™ ***Level 2 Room 2001***
2:30 PM – 3:30 PM *SOFTS03* — ChromeOS* and coreboot* on Intel® Architecture Platforms – A Primer ***Level 2 Room 2006***
4:00 PM – 5:00 PM *SOFTS05* — Intel® Software Guard Extensions Technology Overview and Programming Model ***Level 2 Room 2006***

Wednesday, August 17, 2016

11:00 AM – 12:00 PM *ANATS02* — Apache Spark* in Enterprise Analytics ***Level 2 Room 2002***
11:00 AM – 12:00 PM *IOTTS02* — Building Embedded and IoT Solutions with Intel® Media SDK for Intel® Atom™ Platforms ***Level 2 Room 2008***
11:00 AM – 12:00 PM *SOFTS04* — Enabling Dynamic Usage Models for FPGA with the Accelerator Abstraction Layer Software Technology ***Level 2 Room 2006***
1:15 PM – 2:15 PM *ANATS04* — End-to-End Analytics Solutions with Trusted Analytics Platform ***Level 2 Room 2001***
1:15 PM – 2:15 PM *PCITS03* — Google Play* on Chrome OS* + Intel® Architecture – A Primer on Developing the Best Apps ***Level 2 Room 2000***
1:15 PM – 2:15 PM *SOFTS06* — Machine Learning: Optimizing Deep Learning Usages on Intel® Client Platform ***Level 2 Room 2006***
2:30 PM – 3:30 PM *ANATS06* — The Complete Toolset for Accelerating Analytics – From Optimized System Architecture to Accelerators ***Level 2 Room 2001***
2:30 PM – 3:30 PM *SOFTS07* — Microsoft Azure Stack*: A Platform View – Insights to Hardware Requirement ***Level 2 Room 2006***
4:00 PM – 5:00 PM *ANATS08* — Open Source Solutions for Network Intelligence ***Level 2 Room 2001***
4:00 PM – 5:00 PM *SOFTS08* — Solving the Holy Grail of IoT: “0 Touch” Device Onboarding ***Level 2 Room 2006***

Thursday, August 18, 2016

9:30 AM – 10:30 AM *SOFTS09* — Techniques for Optimizing Cloud-native Runtimes on Intel® Architecture ***Level 2 Room 2006***
10:45 AM – 11:45 AM *SOFTS10* — OpenHPC* and Intel® HPC Orchestrator, System Software Stacks Providing Key Building Blocks of Intel® Scalable System Framework ***Level 2 Room 2006***
1:00 PM – 2:00 PM *CLDTS04* — Cryptography and Compression Acceleration for NFV, Cloud, and Hyper-Converged Solutions ***Level 2 Room 2002***
1:00 PM – 2:00 PM *SOFTS11* — Orchestrating Virtual Security Functions for Software Defined Infrastructure ***Level 2 Room 2006***
2:15 PM – 3:15 PM *PCITS08* — Modern Standby: Why and How ***Level 2 Room 2004***
2:15 PM – 3:15 PM *SOFTS12* — Control Flow Enforcement Technology Targeting Return Oriented Programming (ROP) Attack Prevention ***Level 2 Room 2006***

IDF16
INTEL DEVELOPER FORUM

Legal Notices and Disclaimers

- Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at intel.com.
- Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit http://www.intel.com/performance.
- Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit http://www.intel.com/performance.
- Cost reduction scenarios described are intended as examples of how a given Intel-based product, in the specified circumstances and configurations, may affect future costs and provide cost savings. Circumstances will vary. Intel does not guarantee any costs or cost reduction.
- This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.
- No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.
- Statements in this document that refer to Intel's plans and expectations for the quarter, the year, and the future, are forward-looking statements that involve a number of risks and uncertainties. A detailed discussion of the factors that could affect Intel's results and plans is included in Intel's SEC filings, including the annual report on Form 10-K.
- All products, computer systems, dates and figures specified are preliminary based on current expectations, and are subject to change without notice. The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.
- Intel does not control or audit third-party benchmark data or the web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.

Backup

Robotics in the Curriculum

Embedded

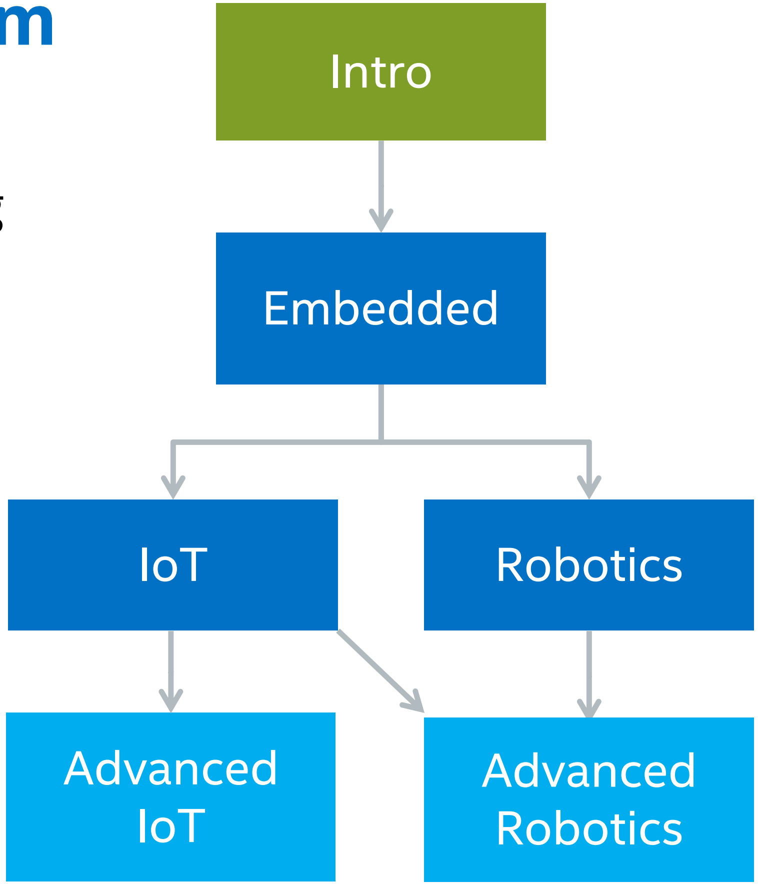
- Low-level hardware interfacing
- RTOS and MCU programming

IoT

- Web and cloud programming
- IoT middleware and patterns

Robotics

- Perception and actuation
- Planning and control



IDF16 INTEL DEVELOPER FORUM