Bridging ROS to Embedded Systems

Morgan Quigley
Open Source Robotics Foundation
Overview

• survey of the diversity of embedded systems and some bridging approaches using ROS

• not intending to argue for/against any particular method for all situations
  – best method is often implied by the application
Motivation

• all robots have embedded systems.
  – it’s not “if” but “how” we talk with them
  – lasers, cameras, motors, radios, etc.
• sometimes, we have control over firmware
• sometimes, we don’t
• many different situations = many approaches
Huge Range of Embedded Systems

Why embed?

• reduce cost, size, weight, power
• more powerful every year
  – massive market forces #embedded >> #pc’s
• real-time requirements are hard/painful to mix with non-realtime code on full systems
  – isolate real-time requirements to embedded
  – more and more code can go in them
Popular Hardware Classes

- **x86**
  - mini-ITX
  - PC/104

- **ARM-A**
  - Android
  - Raspberry Pi
  - Gumstix
  - BeagleBone

- **ARM-M**
  - mbed
  - STM32 Discovery
  - Olimex ARM-USB-TINY-H

- **AVR**
  - Teensy 2.0
  - Arduino

Two Extremes of Integration

“Unified”

“Bridged”

bridge
Unified

- roscpp, rospy
- rosjava
- rosc (future)
- uROSnode
- rosbridge
- rosserial
- custom bridges

Bridged

x86 | ARM-A | ARM-M | AVR
roscpp, rospy, rosjava

• ARM Cortex-A machines can run all of ROS
• Cross-compiling can be tricky. Start with binary distros and/or popular platforms:
  – Android
  – Gumstix (Linaro)
  – Raspberry Pi (Raspbian)
uROSNode and rosc

- ROS node functionality on ARM Cortex-M
- Small size via code generators
- Can chain to other MCU’s
- $1
rosbridge

• JSON protocol to bridge to non-ROS systems
  – for example, connect web browsers to ROS
  – more broadly, connect sockets to ROS

• Much more at http://rosbridge.org and http://www.ros.org/wiki/rosbridge_suite
rosserial

• Bridges selected topics between a ROS system and a serial stream: UART, XBee, etc.
• Ports: Arduino and generic embedded linux
• [http://www.ros.org/wiki/rosserial](http://www.ros.org/wiki/rosserial)
Custom Bridges

- Useful to implement proprietary protocols, unusual data links, complex inter-system communications, etc.
  - hokuyo_node (and 1e6 other drivers)
  - Sandia Hand: https://bitbucket.org/osrf/sandia-hand
Cortex-M cross compiling, debugging

- [http://bitbucket.org/osrf/sandia-hand](http://bitbucket.org/osrf/sandia-hand)
  - downloads a version of arm-gcc
  - clones and builds OpenOCD
  - CMake rules for cross-compiling Cortex-M targets
  - GDB server and client targets, disassembling, etc.
  - custom bootloaders to support UDP, rs485, etc.
Xilinx FPGA simulation

- [http://bitbucket.org/osrf/sandia-hand](http://bitbucket.org/osrf/sandia-hand)
  - Makefile rules for synthesis, download, flash burn
  - Rules for simulation using Icarus Verilog (GPL)
  - Gigabit ethernet emulation to/from simulation
  - Debug comms between simulated FPGA and driver
Acknowledgments and Summary

- ros-sig-embedded
- ROS for Products Workshop

- furious activity at many levels of embedded
- lots of interest in a portable, lightweight ANSI C client for both bare metal and various RTOS’es
  – next talk: rosc