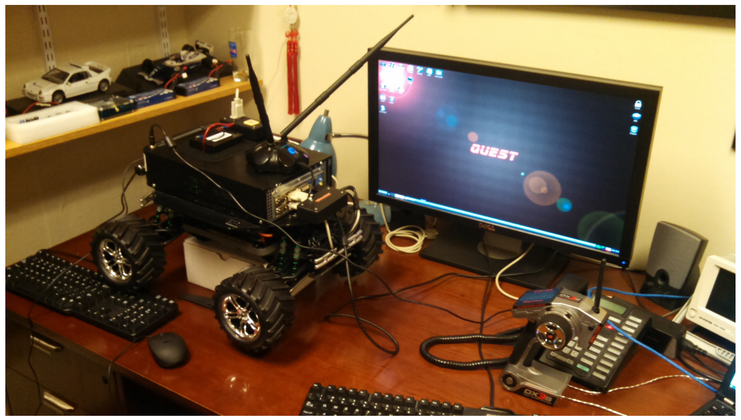
RacerX: High-Speed Autonomous Vehicle Control

Richard West, Ye Li, Zhuoqun Cheng

Motivation

- Build a prototype vehicle to understand the computation and energy demands of autonomous vehicle control
- Simulate real automotive environment to understand the isolation required by subsystems in a vehicle
- Consolidate multiple system components onto a single platform with multi-core and Quest-V separation kernel

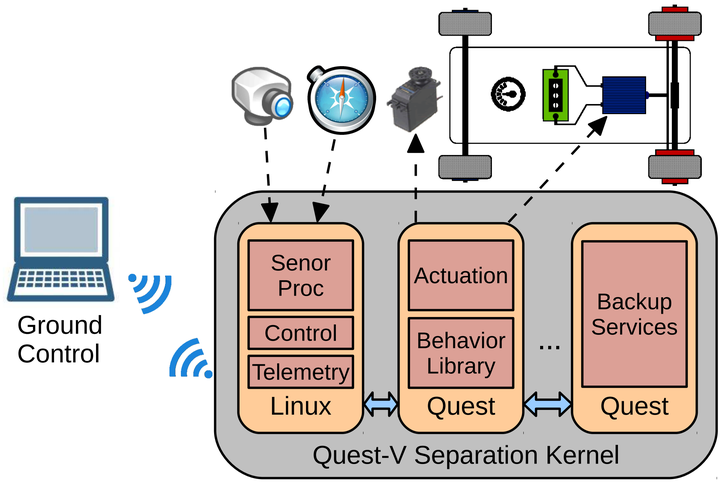
Vehicle Specification



Chassis	• Traxxas E-Maxx (Model 3905) • Modified suspension and gearbox • Final gear ratio: 29.57
Power	• 12V Zeus AGM battery (computer) • 2x 7.2V NiMH battery (drive motors) • M2-ATX mini-box regulator • PicoUPS-120-ATV
On-Board Computer	• Jetway NF9F-H61 Industrial Control Mainboard • 6x RS232 Serial • Intel Core i3-3220T Processor
Sensors	• UM7-LT Attitude and Heading Reference System • 66-Channel LS20031 GPS Receiver Module
Comms	• POWERLINK Hermes 802.11N 300Mbps Wi-Fi • XBee Pro 63mW (Series 2B) • Up to 2km range

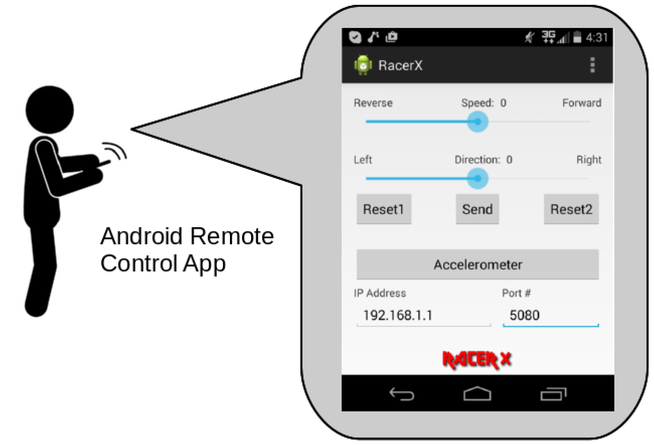
Architecture

- The Quest-V Separation Kernel
  - Uses hardware virtualization to partition resources amongst services of different criticalities.
  - Each partition, or *sandbox*, manages its own CPU, memory, and I/O resources *without hypervisor intervention*.
  - Hypervisor only needed for bootstrapping system + managing commnuication channels between sandboxes.



- Linux sandbox
  - Sensor data processing
  - Sends telemetry to remote ground control station
  - Makes control decisions based on sensor input and sends them to Quest sandbox for vehicle control
- Native Quest sandbox
  - Translates high-level commands from Linux sandbox into low level servo and motor commands
  - Performs real-time servo and motor actuation
- Backup Quest sandbox for fault recovery
- Linux and Quest sandbox communicate via shared memory message passing channel
- Ground control station communicates with the buggy wirelessly using MAVLink protocol and displays telemetry

Demonstration



QGroundControl (Remote PC)

Future work

- Enhance the native Quest sandbox capability
- Implement autonomous decision making logic in the Linux sandbox based on sensor input
- Implement fault detection and recovery mechanisms

Further Info: http://www.cs.bu.edu/fac/richwest/racerx.php

Advisor: Richard West