Leveraging SOA Communication Middleware with TSN for Software Defined Vehicles

ETHERNET & IP @ AUTOMOTIVE TECHNOLOGY DAY
19-20 September 2023 | São Paulo, Brazil

Jakob Zvirchmayr
Product Innovation

Shashank Gaur
Pre-Sales
Typical single-ECU functions
Control loops with strict timing requirements (event-chains)

Requirement A: vehicle stops within x ms at speed of y kph if event z happens
(Sub-)System X needs to detect event z within a ms and trigger reaction of (sub-)system Y within b ms
Zonal Architectures and Software Defined Vehicles

- From single-ECU design to vehicle-wide multi-ECU designs
- Higher integration, no more isolated execution on SoCs/ECUs
- Cover distinct HW platforms, not just variants
- Consolidate and optimize hardware and communication paths
- Continuous updates to SW in a changing HW environment

Functionality from domain controllers shifted towards ECU
Zonal Architecture Evolution

- Up-integration for new architectures: functionality increasingly realized in software
- Enables sharing of resources provided by different ECUs
- Enables optimization of communication flows and compute platforms
- Cloud/Edge connectivity becomes relevant
Software Defined Vehicle

Realizing functionality mainly in Software enables evolution and consolidation of the Hardware platform and legacy systems!

Same SW function Deployed in different physical architectures
Consequences of Zonal Approaches

• Integration of applications from Classic AUTOSAR, POSIX, ...

• Maintain real-time guarantees for communication and FFI for safety-critical applications in a heterogenous and distributed mixed-criticality environment

• Software is continuously upgraded and improved:
  • Embedded Applications & Tooling
  • Increased integration of off-board systems
  • Maintaining SW functionality in a changing hardware environment requires decoupling of SW provided functions from physical deployment = Location Transparency
  • Switching from Signal-oriented to SOA is just one aspect

"with SOA, we got the worst of both worlds ..."
Technology Selection
From Prototype to Series Deployment

- Frameworks like ROS2 are superb for prototyping; typically, such prototype systems must be re-implemented using 'series quality technology'.
- With DDS/Zenoh there exists a clear and defined path from prototype development to series deployment; no switch of design methodology or technology is needed.
Tooling Perspective

- DDS allows to seamlessly transition from loosely coupled to defined topology
- SW Functionality is not at all impacted by the transition
- Integrator and Developer can specify/adapt application QoS (SW Constraints) and SW topology
- Guarantees (Timing, FFI) are configured in a fully automated scheduling step

Series deployment

CC constraint = x ms
Series deployment

CC constraint = x ms

+QoS
+Deployment CFG
+ET (WCET)

(Task/Network) Scheduling Algorithms

CC constraint = x ms

+QoS
+Deployment CFG
+ET (WCET)

(Task/Network) Scheduling Algorithms
Series deployment
Deterministic communication with DDS

Prototype integration of DDS with Task and Network Scheduling shows improved determinism

- Explicit scheduling of tx and rx threads on task level
- Mapping of DDS QoS Parameters to TSN
- Example: Stream isolation properties against low-prio/best-effort traffic
Deterministic communication with DDS

**BEST EFFORT**
- Lost messages
- Delayed execution

**DDS+TSN**
- No lost messages
- Execution delays under control

September 28, 2023
Scheduling Performance

Results

FREE RUNNING

Spikes in latency until data is processed

TASK + NETWORK SCHEDULER

Execution time bound

Load 30% 40% 50% 60%

Load 30% 40% 50% 60%
HQ, Vienna, Austria

office@tttech-auto.com \| www.tttech-auto.com \| +43 1 585 65 38-5000
Assigning tasks/messages to both the time and space domain such that certain constraints are fulfilled, e.g., non-overlap, deadlines, precedences, chains, etc.
Tooling Workflow

1. System Definition
   - Number of Tasks
   - Periods
   - Host Affinity
   - Resources

2. Scheduling Suite Configuration
   - Timing Budget
   - Core Affinity
   - Jitter
   - Category (Time/Event/Data Driven)
   - Free Running App
   - Computational Chains

3. Task Mapping & Comp. Chain Scheduling Tool

4. Network Scheduling Tool

5. Task Schedule Visualization

Network Schedule Configuration
- AUTOSAR
- RTOS
- Linux

Network Schedule Configuration

ECU

Switch