<table>
<thead>
<tr>
<th></th>
<th>Agenda</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Automotive Switches: Software and Hardware Architectures</td>
</tr>
<tr>
<td>2.</td>
<td>Automotive Switches and AUTOSAR</td>
</tr>
<tr>
<td>3.</td>
<td>Smart Switches and AUTOSAR</td>
</tr>
<tr>
<td>4.</td>
<td>Outlook</td>
</tr>
</tbody>
</table>
Evolution of In-Vehicle Ethernet Architectures

- Isolated point-to-point networks for dedicated use cases with increased bandwidth demand

- Introduction of Ethernet switches and scaling the Ethernet network
Scaling Ethernet and tackle wiring complexity with zonal architectures

Mindset: Ethernet is a shared, “cross domain” communication medium

Redundancy on Ethernet will require enhanced TSN features to be supported by all switches
Evolution of Ethernet Communication Requirements

- Requirements changed over time
  - from the “sheer need for more bandwidth”
  - to a reliable and secure networking in-vehicle backbone

- flexible / dynamic service-oriented communication exists in parallel
- with control and streaming traffic with "TSN" requirements on the same Ethernet network
Automotive Switches: Software and Hardware Architectures

Challenges

- **Switches are essential** for Ethernet networks
  - enable communication between multiple Ethernet nodes
  - control network access, latency and bandwidth
- **Switches play a central role** to deploy
  - time synchronization and TSN features
  - Network reliability and - security

- The number of switches in-vehicle is increasing
- The feature set to be displayed is increasing as well
  - *Increased configuration and qualification effort*

? Open questions
- How does the HW and SW architecture look like?
- Which SW features will be necessary on the switch?
- How do we ensure overall configuration consistency?
- How is switch SW updated?
Typically, a **switch is a subsystem of an ECU** attached to an MCU on the same PCB

“Traditional” AUTOSAR approach:
- MCU manages the switch including its PHYs as **external peripherals** of the MCU
- using AUTOSAR drivers
- via **management interfaces** e.g. MDIO or SPI
Limit of MCU managed Switch Approach

- The MCU based switch management **does not scale** for some use cases
  - e.g. handling of switches with many ports for PTP residence time compensation
- The MCU based switch management is **not feasible at all** for specific features
  - e.g. firewalling via an attached MCU will not provide the necessary throughput

**Limits of MCU managed Switch Approach**

- Automotive Switches and AUTOSAR
Today’s switch devices come along with an **own CPU subsystem** and allow – to some extent – an **independent operation** of the switch.

Switch devices with an **own CPU** can be used as a “smart” switch subsystem of the MCU and take over relevant network-specific tasks.

**Offloading** of networking tasks from the MCU to the switch.

Other interesting use cases come to play...
Following approaches are common:

1. “Downgrade the smart switch to a dumb peripheral” and run **AUTOSAR switch driver on MCU**
   - Perquisite: switch *must not* execute own firmware to avoid conflicting register access
   - Drawback: the switch resources cannot be used → this is neither smart nor sustainable

2. Alternatively, the **switch driver is removed** on the MCU; switch executes **proprietary firmware**
   - Drawback: Switch is **decoupled from the AUTOSAR** configuration and update workflow and MCU is not “switch aware” anymore
   - Questions left open to be solved in project scope:
     > How do we achieve configuration consistency?
     > How does the software update and UDS diagnostics concept look like?
     > What about automotive specific protocols or extensions to be supported on the switch subsystem?
The switch **ECU system** consists of an

**MCU** typically running **AUTOSAR** based software and a connected

**Switch** with an **integrated CPU** which could run own software too
Switch ECU: Multi-Processor System with distributed Feature Set II

- Vision: distributed software architecture
  - MCU and switch CPU
  - Quick **startup** / **wakeup** (incl. TC10)
  - Basic **switch configuration**
  - GTS (PTP)

- **Software- and configuration update**
- UDS diagnostics

- Network management (**PNC**)
- AVB / TSN features

- **Firewalling**
  - Other use cases / OEM specific software
    - *e.g. variant provisioning*
  - Close future: **MACsec**

Vision: distributed software architecture
MCU and switch CPU
- Quick **startup** / **wakeup** (incl. TC10)
- Basic **switch configuration**
- GTS (PTP)

- **Software- and configuration update**
- UDS diagnostics

- Network management (**PNC**)
- AVB / TSN features

- **Firewalling**
  - Other use cases / OEM specific software
    - *e.g. variant provisioning*
  - Close future: **MACsec**
What kind of software do we need on the switch?
Which pieces are meaningful to be offloaded from the MCU to the switch subsystem?
Switch-local OS and hardware abstraction layer are perquisites

Relevant functional clusters are
- Switch initialization and basic configuration
- PTP handling and further TSN features
- Extensions to allow a software update and diagnostics via the MCU
- Network management
- Security features
SMART SWITCHES AND AUTOSAR

Switch ECU: AUTOSAR aware Switch Software – minimal Configuration

- AUTOSAR aware switch software running on the switch allows a standalone boot and operation.
- Porting of further standard AUTOSAR software to the switch subsystem can be done with small effort.
- Configuration workflow and tooling is identical to the AUTOSAR stack running on the attached MCU.
- Example of automotive specific implementation: PTP (IEEE 802.1AS) with AUTOSAR extensions covered with EthTsyn module.
Today’s switches already allow far more extensions than just a switch-local PTP handling.

With an AUTOSAR aware smart switch solution we can get the best out of smart Ethernet switches:

- add further production approved software available in the AUTOSAR eco system e.g. for network management or diagnostics
- increase the overall system performance
- shorten the time to marked with qualified software and a proven architecture
- gain time with one solution, tooling and workflow

Switch ECU: AUTOSAR aware Switch Software – A Scalable Approach
Challenges To be Overcome

- The software executed on the switch today is limited by available hardware resources
  - Future use cases e.g., firewalling and IDS will require significant resources
  - Features like MACsec will require a switch-local key storage and crypto an acceleration

- Mindset change in the automotive industry necessary:
  - **Switches are complex subsystems** rather than “simple peripherals”
  - Switches are active components with own communication needs
    - Hence, switches require own MAC -, IP - and even diagnostic addresses
  - Software share and feature set on switches will play an important role in future ECU projects to be considered by the ECU “component responsible” on OEM side
  - Transport formats and workflows need to be discussed
    - AUTOSAR XML may be a solution or at least a basis for many OEMs
Outlook

Outlook: Future Extensions in Smart Switches

- Software share on switches will increase
- Integrated software architecture is needed to meet the time to market – AUTOSAR may be a solution

- Next:
  - Stringent functional safety concepts will be required for automated driving
  - Introduction of MACsec and further TSN features is on the way

- Already more than a "vague vision": **The switch as a fully independent single chip device!**
Your questions are welcome!
Please visit us also at our booth.

www.vector.com

Author:
Markus HELMLING
Vector Germany