ADAS development – Introduction

System overview

“Sense” → Sensor ECUs – single or many

- LIDAR
- RADAR (SR & LR)
- Ultrasound
- Front Camera
- Side & Reverse (camera/ radar / ultrasound)

“Display & (Re)Act” → Signaling & Actuation system ECUs
- Passive: HMI (Cluster)
- Active: Engine & Transmission, Brakes, Steering

“Evaluate & Decide” → Data Fusion + ADAS Logic
- Integrated in sensor ECU(s), or
- Dedicated ADAS / autonomous driving processor (hi spec / multicore)

“Virtual” sensors
- Maps (+GPS)
- Car2X
- Road conditions
- …
Example scenario - Fusion of multiple sensors

Why?
- Field of view & range of single sensor limited
- Highly reliable environmental model (object positions, sizes, speeds) needed as basis for reaction/driving maneuver planning

How?
- Combine information from multiple sensors to “get a better understanding” of everything around the vehicle

Example: Detection of occluded objects by fusing vision and Car2X (position) information – to better calculate safe vehicle paths
Vector and BASELABS partner for ADAS development tools

- In August 2014 Vector and BASELABS formed a strategic partnership for driver assistance system and automated driving in the automotive field (ADAS systems)
- In future, Vector focusses on tools for the development of ADAS and autonomous systems
- In future, BASELABS focusses on software for data fusion in multiple-sensor scenarios including algorithms and application support in the ADAS and automated driving field
- Together, BASELABS and Vector will cooperate closely to provide a complete and integrated solution consisting of tools, algorithm libraries, services and consulting to our customers. This combined offer allows efficient and convenient development of state-of-the-art ADAS and autonomous systems.
Contents of Vector ADAS solution / Typical Use Cases

ADAS development – Introduction

- ADAS ECU road validation
- ADAS Algorithm & software development
- Rapid prototyping and Bypassing
- High speed ECU (RAM) measurement

vADASdeveloper
Development tool for implementing, debugging and testing multi sensor applications in Visual Studio (C#, C/C++)

BASELABS | Create
Algorithm toolbox / library for sensor data fusion, probabilistic filtering and tracking

BASELABS | Modules
Customer-specific ADAS software modules for prototyping and test
vADASdeveloper

Visual Studio Integration

“Designer”: Drag & Drop data flow / system design

“Toolbox”: Sensor and component library, easily expandable.

Direct application source code access
## ADAS solution – Overview products

### Visual Studio Integration

<table>
<thead>
<tr>
<th>Tool</th>
<th>Data fusion</th>
<th>Code generation</th>
</tr>
</thead>
<tbody>
<tr>
<td>vADASdeveloper</td>
<td>Templates for e.g. multiple object tracking</td>
<td>GUI for code generation</td>
</tr>
<tr>
<td>GUI for drag &amp; drop system design</td>
<td>User support for algorithm development</td>
<td></td>
</tr>
<tr>
<td>Visualization of data e.g. birds eye view</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### IDE integration

- (Microsoft Visual Studio)
- IDE integration
- GUI for drag & drop system design
- Visualization of data e.g. birds eye view
- Templates for e.g. multiple object tracking
- User support for algorithm development
- GUI for code generation

### IDE-independent libraries

- BASELABS Modules
- Matlab / ADTF / CANoe / CANape / ...

### BASELABS | Modules

- SDK
- Operating system Drivers
- Math libraries
- SDK

### BASELABS | Code

- SDK
“Wizards” to support core fusion algorithm selection & parameterization
Data Recording / Logging

Support for arbitrary sensors.
Easy integration of new sensors

- By user
- Offered as service by BASELABS

Parallel recording capability for high data load.

Reliable time-stamping and synchronization.
Example: Data Fusion

Data Fusion algorithm
- Fast development with CREATE
- Re-Use existing (C/C++) code

Ready-to-use sensors like Ibeo Lux

Convenient integration of own / proprietary components
- Own sensor drivers
- Own source blocks for e.g. custom file formats
Example: Multisensor Application (radar/video data fusion)
Contents of Vector ADAS solution / Typical Use Cases

- **ADAS ECU road validation**
- **ADAS Algorithm & software development**
- **Rapid prototyping and Bypassing**

**High speed ECU (RAM) measurement**

- VX1000 for High Speed RAM access
Measurement bandwidth requirements

Example:
- Chassis 40 MB/s
- Powertrain 5 MB/s

Example:
- EV/HEV 10 µs
- ADAS DCU 40 MB/s
- Radar ECU 40 MB/s + 100 MB/s Raw data

Example: 1 MB/s
- Damping, Steering
- LDW, Parkassist

CCP/XCP on CAN
Data rate

VX1000 - High Speed RAM access
VX1000 - High Speed RAM access

Standard Debug / Trace Ports:
- JTAG / DAP
- NEXUS / AURORA

Automotive qualified:
- High Speed Cable

Standard ASAM Interface:
- XCPonEthernet

ECU \rightarrow Plug On Device \rightarrow Base Module \rightarrow Measurement Tool

10 cm \rightarrow \text{max.} \rightarrow 7,5 m \rightarrow \text{max.} \rightarrow Ethernet

- Measurement principles (depending on muC):
  - RAM data copy: up to 3 MByte/sec
  - Data Trace: up to 40 MByte/sec \textit{without} μC load

- Rapid Prototyping:
  - Very low latency bypassing

- Bus Monitoring:
  - 4 x CAN / 1 x FlexRay channels
**Contents of Vector ADAS solution / Typical Use Cases**

**ADAS ECU road validation**

ADAS Algorithm & software development

Rapid prototyping and Bypassing

High speed ECU (RAM) measurement

---

**CANape Option „Driver Assistance“**

Time synchronized measurement and calibration of ECUs.

Graphical object overlay
- Video/GPS window
- Bird’s eye view
Synchronized Recording of CAN, XCP,... and Video Signals

- Time synchronized recording of measurement signals and video/audio
- Support of all DirectX/DirectShow cameras (up to 240 fps)
- Support of compressed video capture
- Triggered recording with pre and post time – also for key-frame based video formats (MDF hosts time sync info; video in linked AVI)
- Time offset between measurement signals and video pictures adjustable offline evaluation
Object overlay

- Graphical object overlay for detected objects like vehicles, lane markers, traffic signs, ... 
- Freely configurable birds-eye view 
- Flexible and easy to use Object Editor 
- Fast camera calibration
Use Case: Autonomous Parking

- Display of
  - car movement
  - steering angle
  - ultrasonic sensor
  - covered street surface

- Measurement:
  - Object distance
  - Object angle

Angle: 45°57’10.7572’’ (ext)
Use Case: Radar ECU

- Bird’s-Eye View
- GPS View
Use Case: Lane Departure Warning
Screenshot for complete Automotive Setup

- **Simulink Model Viewer**
- **Signal display**
- **Calibration Parameter**
- **GPS position drawing**

- **Diagnostic**
- **CAN, LIN, FR, Eth. / CCP, XCP Trace**
- **Video + Overlay**
Summary

- Vector offers a complete ADAS tooling environment for all kinds of ADAS use cases:
  - ADAS software / data fusion algorithm development
  - Software in the Loop
  - Hardware in the Loop
  - Rapid Prototyping / Bypassing
  - Road Validation and Logging
Thank you for your interest.

Your questions?

Author:
Alexander Aydt
Vector Informatik GmbH