MPC3 Sys Arch and 1V Application Process

YAO Xingkai

10/17/2022

BOSCH

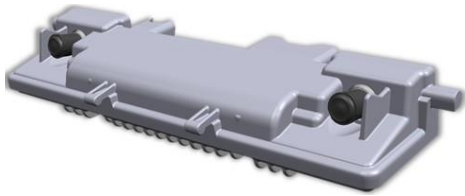
MPC3 Sys Arch and 1V Application Process

Topics

- MPC3 System Architecture
- MPC3 Software Architecture
- 1V Application Process
- Q&A

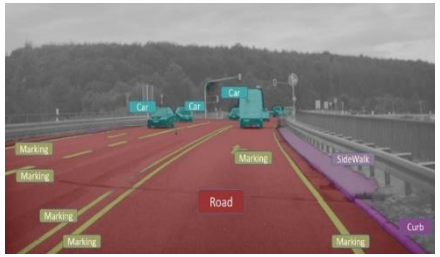
MPC3 Sys Introduction

MPC3 Main Pillars for Success



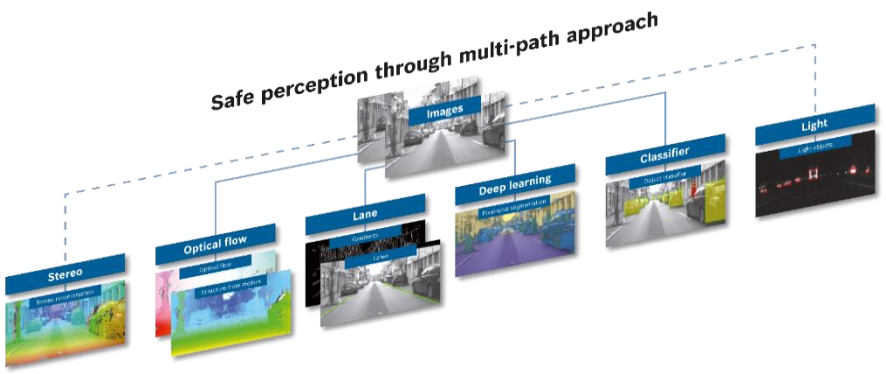
New and cutting edge algorithms

Deep Learning based
semantic segmentation

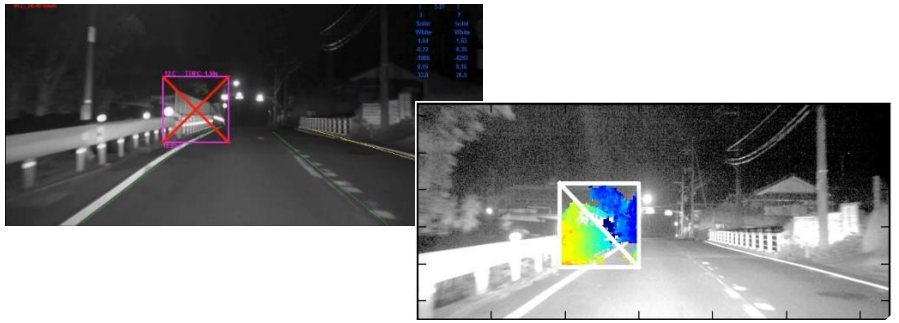


Dense Optical Flow **IP for generic object detection**

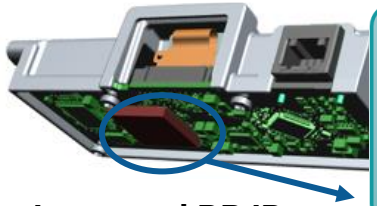
Multi-path approach

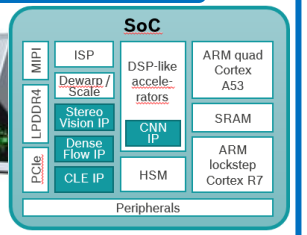


Combination of AI methods & classic computer vision techniques.



High performance Bosch SoC





- **Integrated RB IP**
- latest 16nm technology, max performance with highest **power efficiency**
- SoC defines state of the art **safety/security**
- Positive market feedback:
  Chip already **sold to other Tier1**
  (license RB-IP in open market mode)

Market roll-out

Customers acquired

JAGUAR · LAND ROVER · HONDA · BYTON · 北汽新能源 BAIC BJEV · FAW · 吉利汽车 GEELY AUTO · 吉奥汽车 GONOW · WELTMEISTER 威马汽车 · XPENG · 长安汽车

BOSCH

MPC3 Sys Arch and 1V Application Process

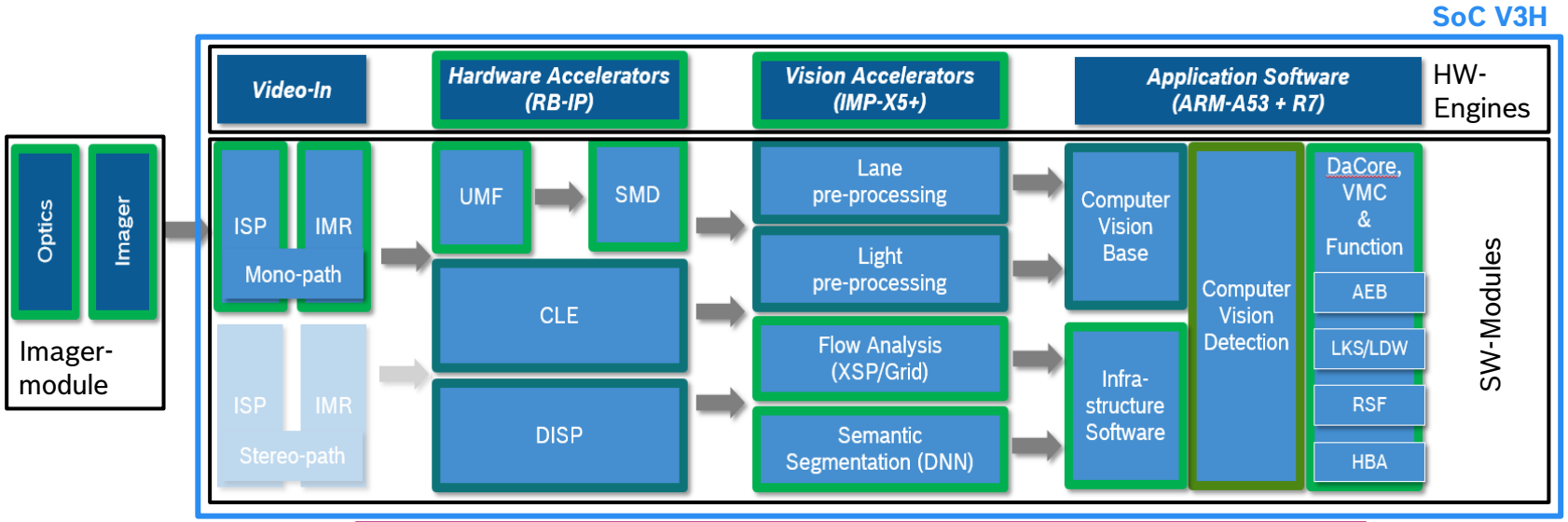
Topics

- > ***MPC3 System Architecture***
- > MPC3 Software Architecture
- > vACC based on ACF
- > Q&A

BOSCH

MPC3 System Architecture

Signal Chain:



Complexity:

- 2,1 MP, 45Hz => 185 MB/s input-stream
- ~25 processing cores on SoC, full parallel and asynchronous calculation
- 87 independent tasks on µC
- 4,3 Mio lines of code
- 1000 SW-builds per day
- >130 SW-Module

Multi Purpose Camera:

- 4 main functions streams:
- Video Out, VRS

longitudinal control (AEB, ACC)

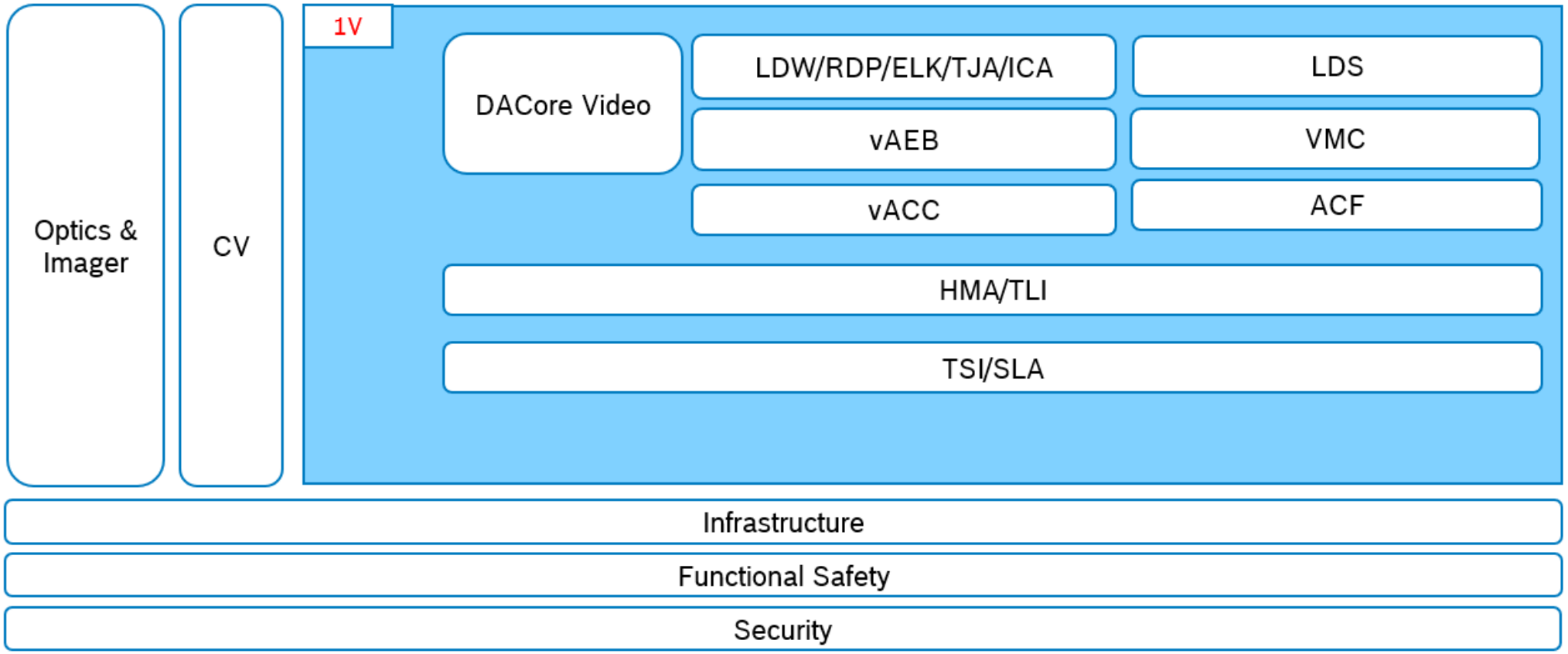
light function (HBA, Matrix)

Road sign information (RSR, RSF)

lateral control (LDW, LKS)

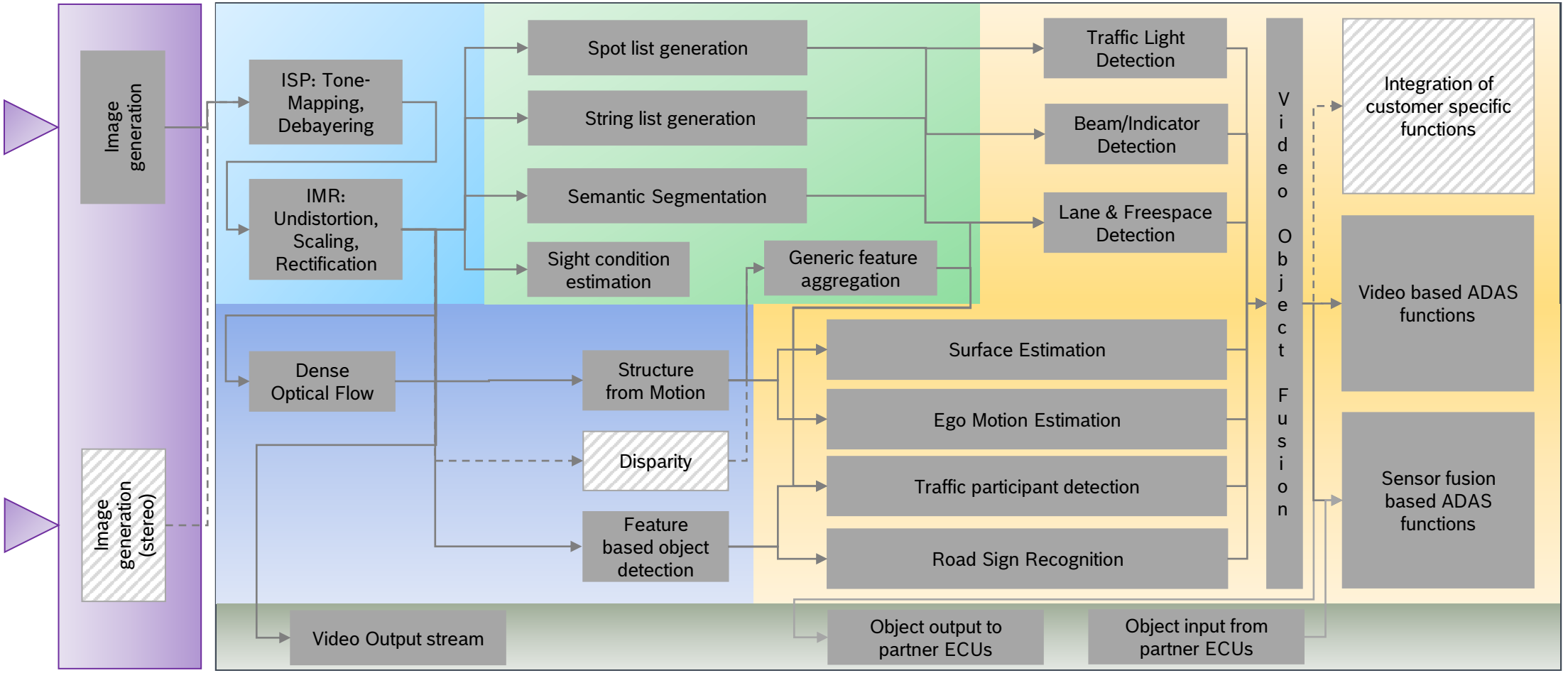
Radical and "full" change: SoC, Imager, Optics, SW-architecture, algorithm, tooling, ...

MPC3 System Architecture



MPC3 System Architecture

Algorithm Elements and Distribution



MPC3 System Architecture

Major Processing Unit

- **ARM Cortex R7 DualCore Lockstep running on 800 MHz**
- **Software will be designed for**
  - Most parts of Basic Software
    - Common UBK Basic Software (CUBAS)
    - CUBAS adapter to middleware
  - Master of „standard peripherals“
  - Safety Controller (Safe Island)
    - Hardware lockstep implementation
    - Only basic algorithmic

- **Limitations**
  - No execute from Flash (because of EEPROM Emulation)
  - No L2 Cache, but 32kB dTCM and 32kb iTCM
  - Must run on 800kB internal RAM (200kB used by SPU)
  - Communication only via middleware to APU Application (shared memory)
  - Safety Level: ASIL B/C
  - No direct access to Flash memory (only via security controller)

- **ARM Cortex A53 QuadCore running on 1000 MHz**
  - 64-bit ARMv8-A Architecture
- **Software will be designed for**
  - Video specific low lever driver
    - ISP, IMR, BoschIP, ...
  - Image proccessing algorithmic
  - Computer Vision stack
  - DA-Core / Functions

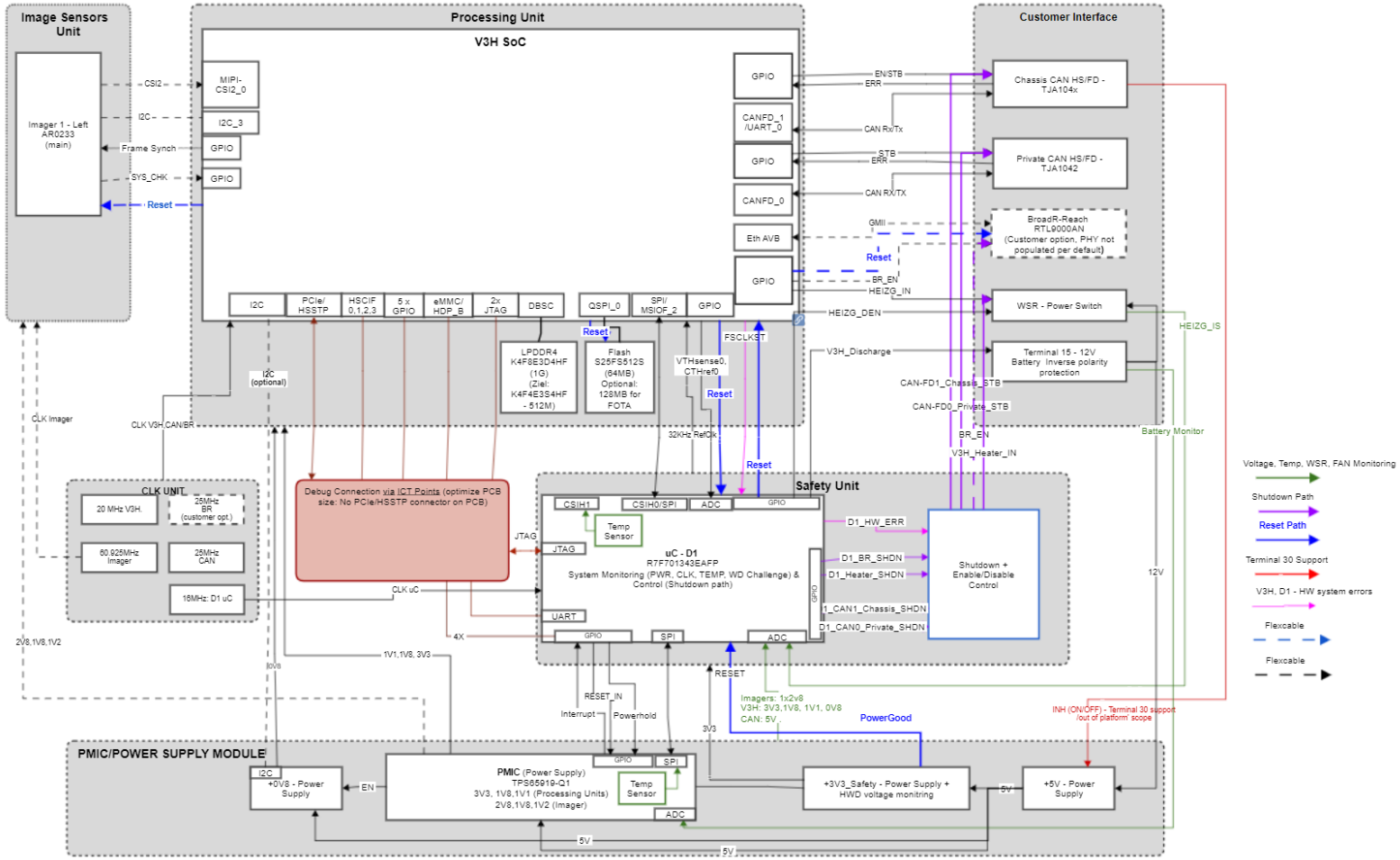
- **Limitations**
  - Asymmetric multiprocessing (AMP): One OS, but static tasks for cores
  - Runs from external DDR only
  - Communication only via middleware to RPU Application (shared memory)
  - Safety Level: ASIL B
  - Powered off in degradation: High temperature, Safety, ...

- **Renesas RH850 / ICU-MX**
- **Software will be designed for**
  - Security only
- **Limitations**
  - Must fit into 200kB internal RAM
  - Master of the external Flash

MPC3 System Architecture

Detailed HW Block Diagram – MPC3 PCB

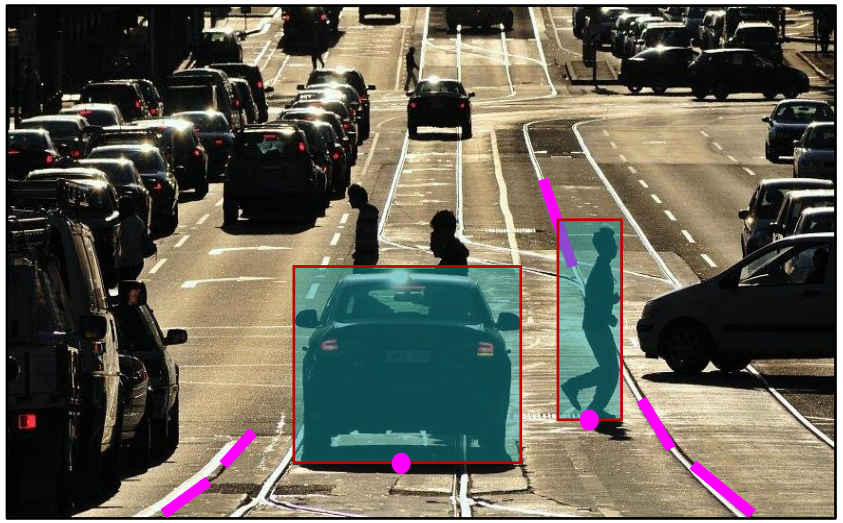
SoC	4 x ARM Cortex A53 Safe Island: R7 Lockstep Hardware-Security-Module(HSM) ISP, Rectification-Module CV DSP Engines Bosch IP 16nm FFC TSMC
RAM	1x 1GB (512MB) LPDDR4 @ 3200MT w/ ECC interleaved
ROM	NOR via QSPI 64MB
Power supply	Size & eff. optimized; Voltage monitoring; Challenge-Response WD
Functional Safety	Safe Island: core lockstep (up to ASIL-C); Other: periodic HW testing (up to ASIL-B)
Image Output	LVDS or encoding via integrated H.264 enc 10bit@1080p@30fps as option
Communication	• 2 x CAN FD • 1 x Ethernet • Flexray as option
Security	•HSM module; secure boot, authenticated and encrypted messaging, symmetric & assymetric keys, …
Boot	• 250ms until 1st CAN default message (including safe & secure b oot) • 3s until algorithms are running

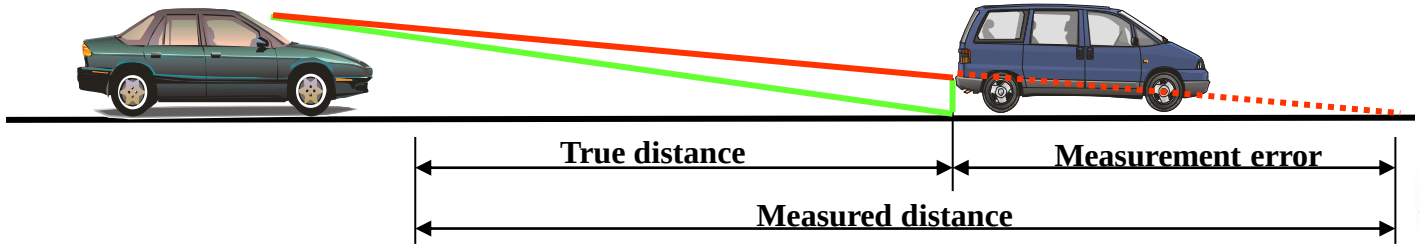


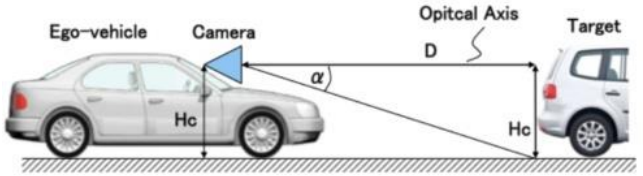
BOSCH

MPC distance estimation（option1）

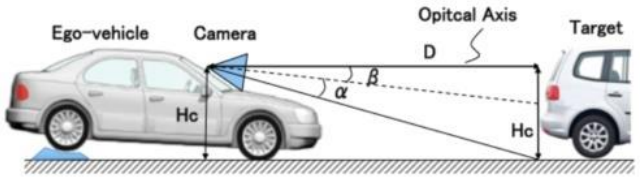
- Distance estimation by glancing intersection
- Necessary inputs:
  - (Dynamic) pitch angle
  - Surface model
  - Footpoint of vehicle bounding box





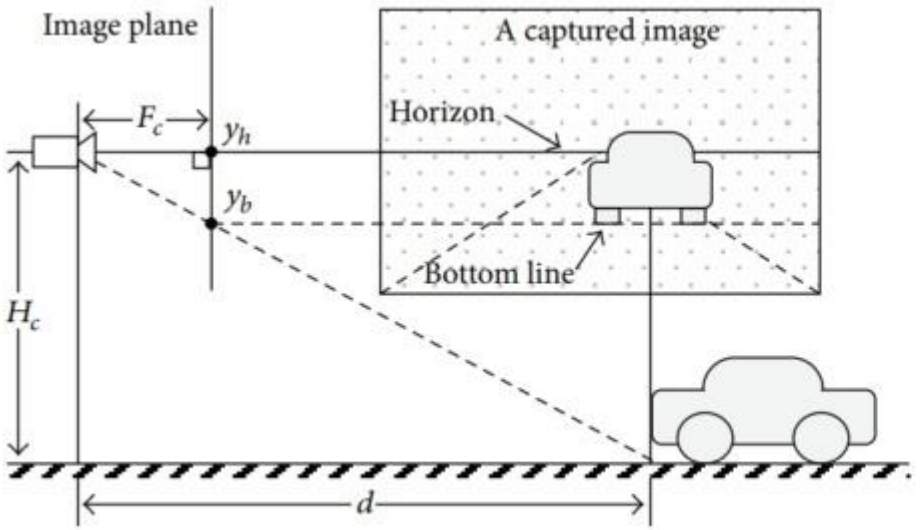


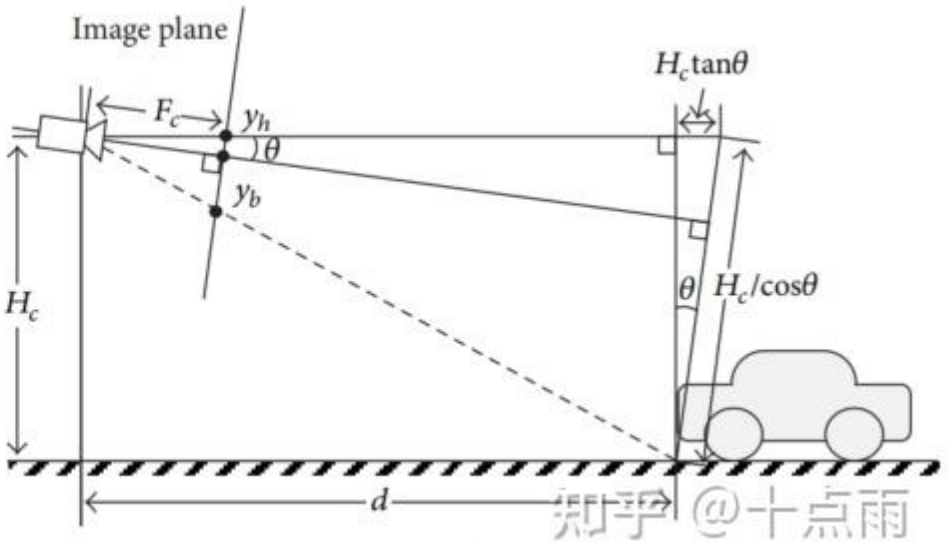
(a) Pitch angle not exist



(b) Pitch angle exist

Mpc distance estimation（option1）





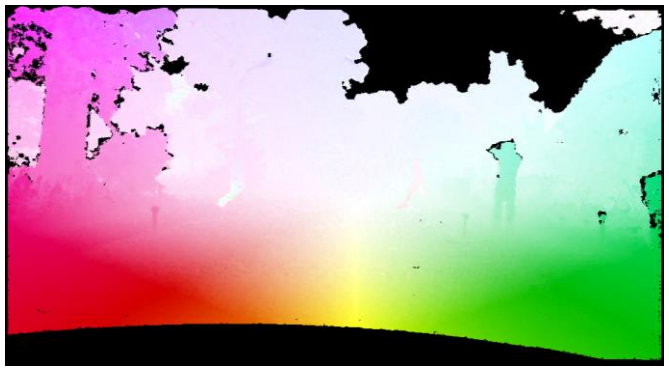
$$d = \frac{F_c \cdot H_c}{y_b - y_h}$$

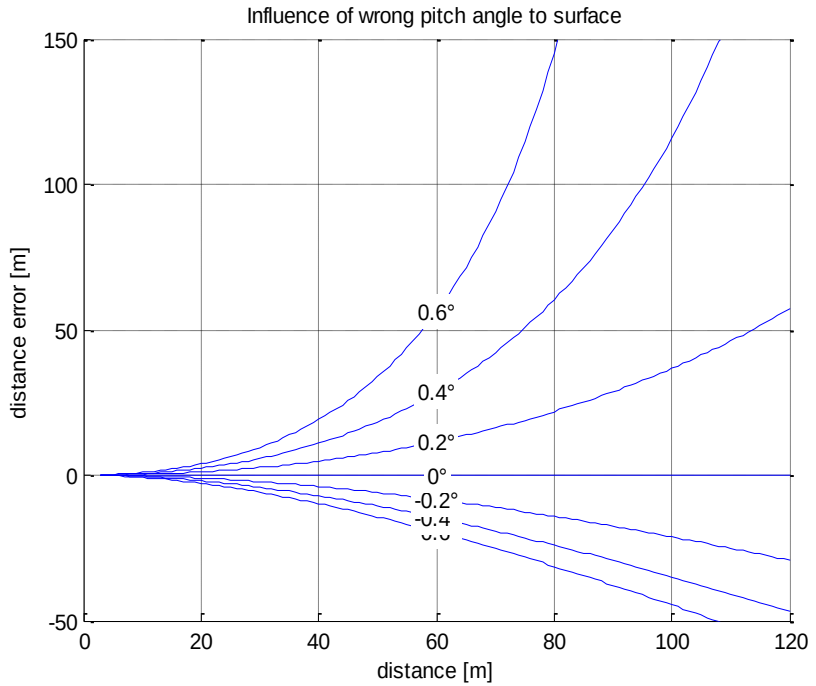
$$d = \frac{1}{\cos^2\theta} \cdot \frac{F_c \cdot H_c}{y_b - y_h} - H_c \tan\theta$$

Reference: https://zhuanlan.zhihu.com/p/419816311

Influencing factors on distance estimation

- Caused by glancing intersection, small errors result in large errors in the distance estimate
- Influencing factors
  - Pitch angle (calibration and ego-motion)
  - Bottom edge of bounding box in the image
  - Surface estimate
    - Main error source in Gen2
    - Optimized on optical flow in Gen3

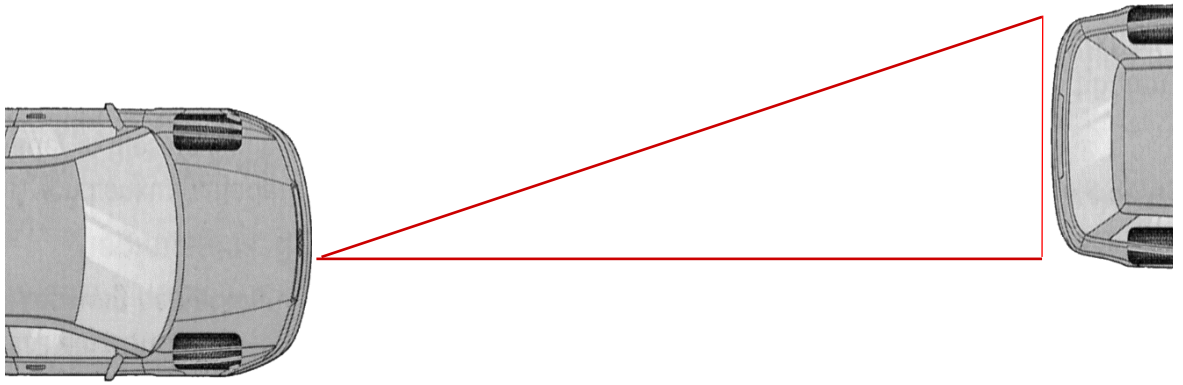




Distance estimation (option 2)

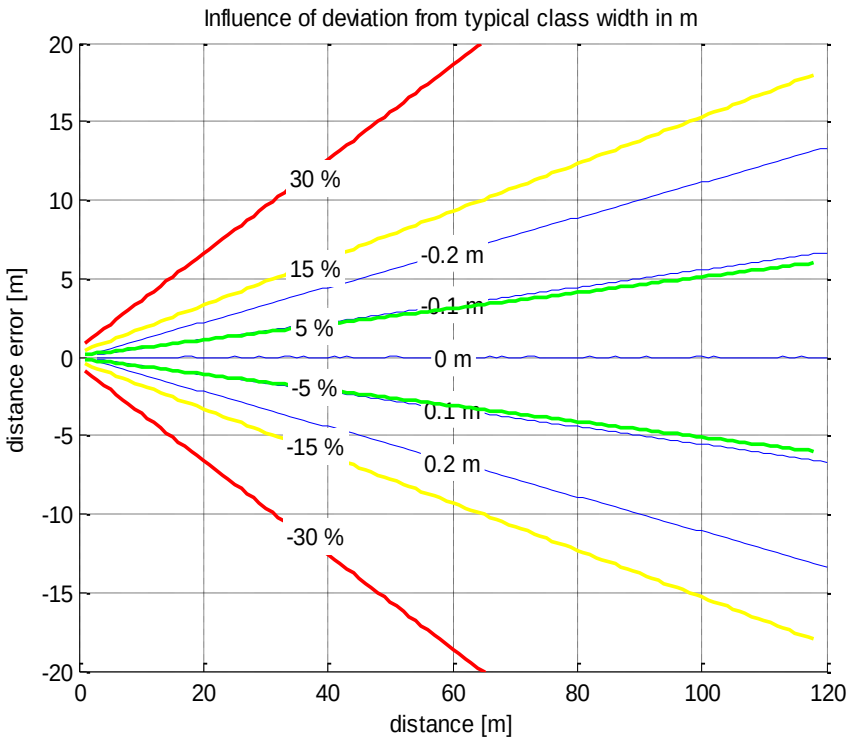
- Estimate using measured width in the image

- Prerequisite: Assumption on class-specific widths, e.g. 1.80m for cars and 2.40m for trucks
  - Deviations from typical object class widths cause distance errors



Distance estimation (option 2)

- Deviations from typical object class widths cause distance errors
- Examples:
  - Golf 1.735
  - Corsa 1.646
  - Q7: 1.983
  - SL500 1.820
  - Toyota hilux 1.760–1.835
- Similar errors are caused if width estimation in image is erroneous



Summary distance estimation

- Distance estimation with mono video possible using
  - surface model and pitch angle estimation and/or
  - vehicle model assumptions

- In practice: Combine both approaches to minimize error

- Rule of thumb: The closer the object, the more reliable the estimate with option 1.

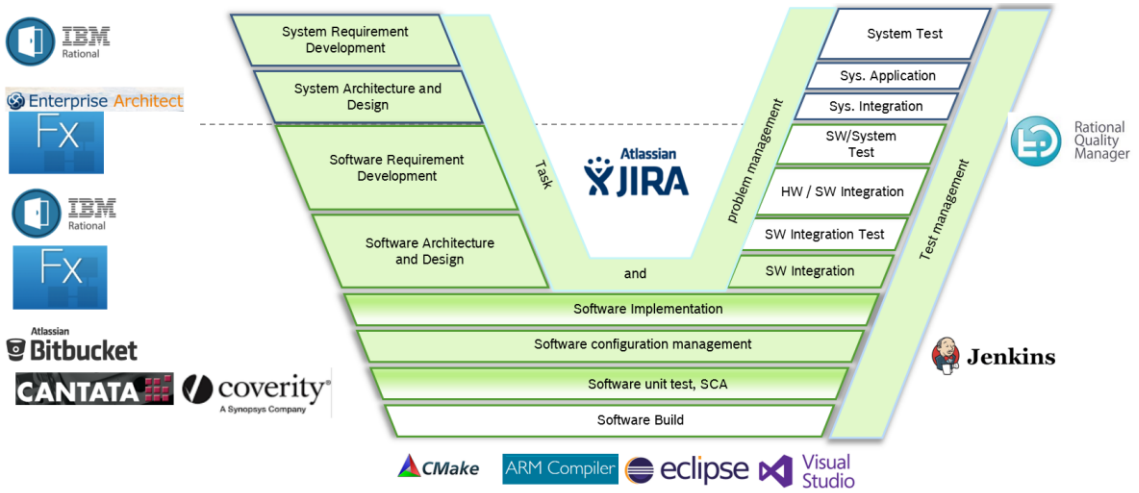
MPC3 Sys Introduction

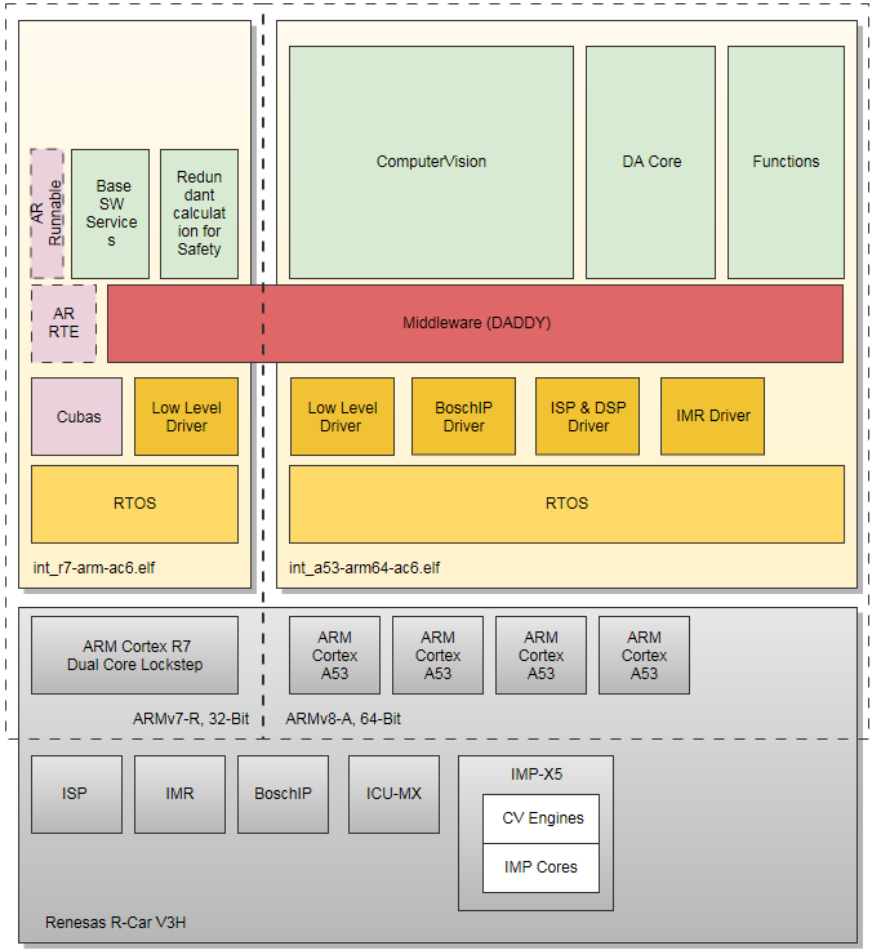
Topics

- MPC3 System Architecture
- ***MPC3 Software Architecture***
- 1V Application Process
- Q&A

MPC3 Software Architecture

Overall SW Information



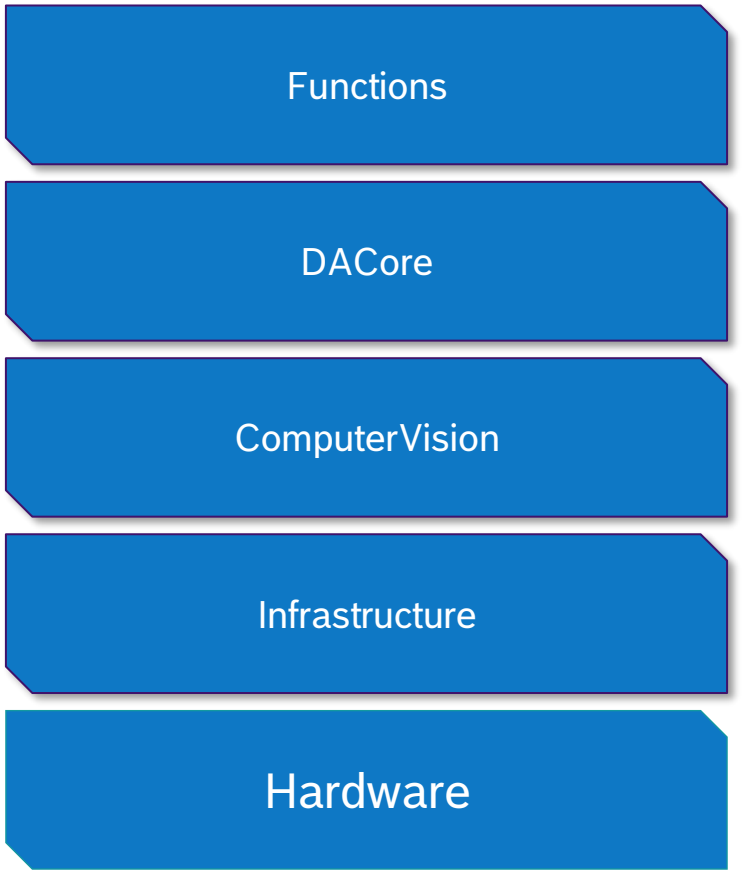


- ***Heterogeneous Architecture***
  - Renesas RH850 / ICU-MX
  - ARM Cortex R7 runs on 32-bit ARMv7-R Architecture
  - ARM Cortex A53 runs on 64-bit ARMv8-A Architecture

MPC3 Software Architecture

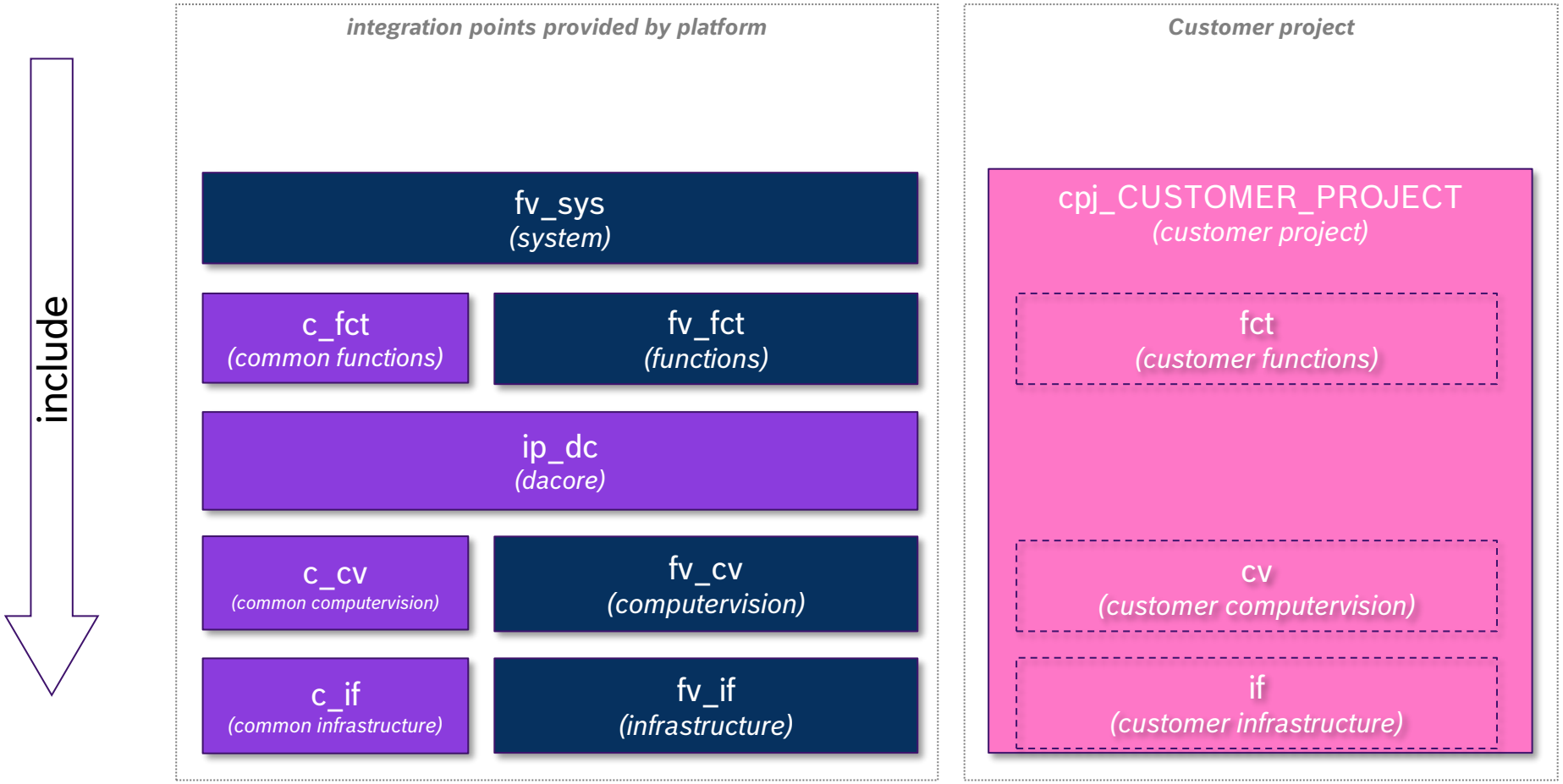
Overall Software Architecture

- Functions:
  - Controlling the behaviour of the vehicle
- DACore:
  - Driver Assistence Core: Situation Analysis, Perception, ...
- ComputerVision:
  - Algorithmic calculation
- Infrastructure:
  - Basic Software, Hardware Driver, Operating System
- Hardware



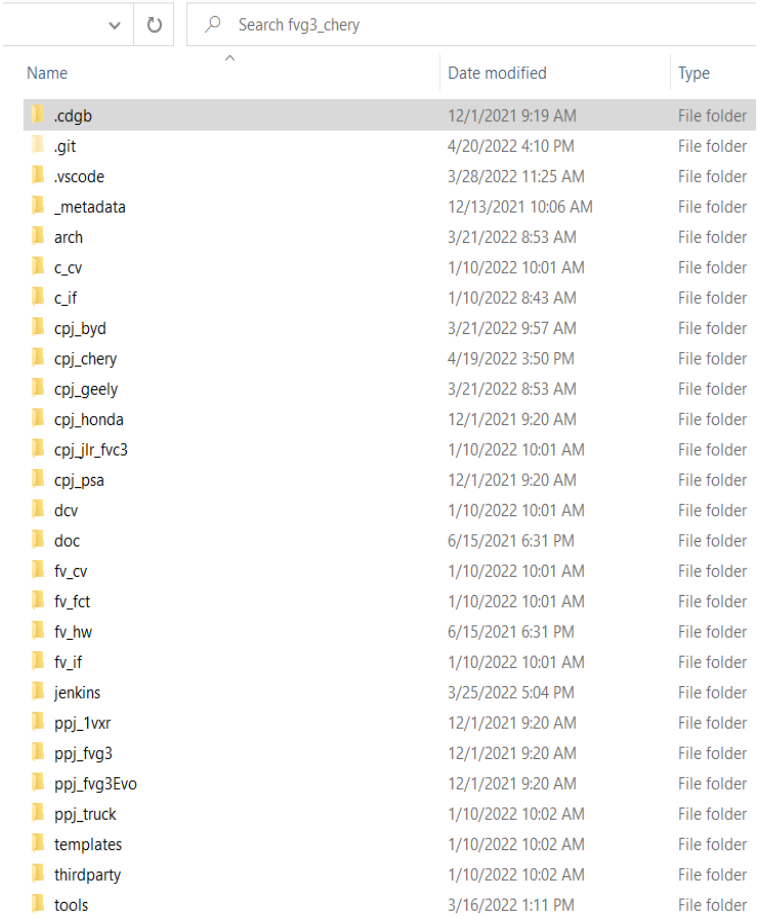
MPC3 Software Architecture

Hierarchical structre



MPC3 Software Architecture

Folder structure



Project Folder	Content
_metadata	Metadata
arch	Toplevel architecture folder (e.g. overall software requirements)
c_cv	Common Computer Vision (CV) components, which are shared between different products lines (e.g. NRCS and FV)
c_if	Common Infrastructure (IF) components, which are shared between different products lines (e.g. NRCS and FV)
cpj_xxx	Customer project implementations (open variations and customer specific components)
dcv	Integration point for DaCore
doc	Additional documents
fv_cv	Platform computer vision integration point for front video
fv_fct	Platform functions integration point for front video (preliminary)
fv_hw	Platform hardware integration point for front video
fv_if	Platform infrastructure integration point for front video
jenkins	Jenkins related files
ppj_xxx	Platform project implementations (open variations and platform project specific components)
templates	Templates (e.g. component template)
thirdparty	Third party content
tools	Tools (e.g. cantata, less,...)

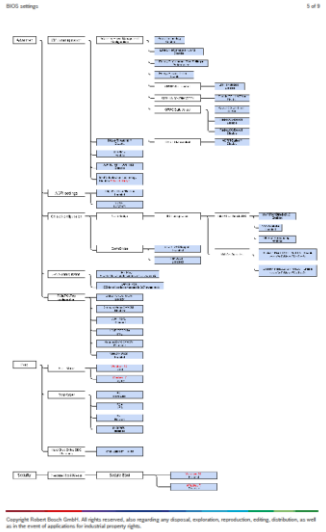
MPC3 Sys Introduction

Topics

- MPC3 System Architecture
- MPC3 Software Architecture
- ***1V Application Process***
- Q&A

1V Tool Chain

➢ ADTF User guideline



ADTF 2.14.2 安装

1V Tool Chain

➢ How to extract CAN data from MF4

preconditions: ADTF has been installed on your personal PC

Run cmd in below path: C:\TOOLS\COMMON\MDF\exe\x64, write command: mdf -input=D:\08717C_MPC3_20220323_094444_001.mf4 -name=CAN_00 -output=stream_data.asc asc

```
C:\TOOLS\common\MDF\ADTF_2.13.3\exe\x64>mdf -input=C:\Users\YXACSZH\Desktop\Chery_8717C_20220112_100513_003.mf4 -name=CAN_00 -output=stre
am_data.asc asc
Opening file: C:\Users\YXACSZH\Desktop\Chery_8717C_20220112_100513_003.mf4
Exporting .asc file(s)
Exported stream CAN_00 to C:\TOOLS\common\MDF\ADTF_2.13.3\exe\x64\stream_data_CAN_00.asc

C:\TOOLS\common\MDF\ADTF_2.13.3\exe\x64>mdf -input=C:\Users\YXACSZH\Desktop\Chery_8717C_20220112_162010_005.mf4 -name=CAN_00 -output=stre
am_data.asc asc
Opening file: C:\Users\YXACSZH\Desktop\Chery_8717C_20220112_162010_005.mf4
Exporting .asc file(s)
Exported stream CAN_00 to C:\TOOLS\common\MDF\ADTF_2.13.3\exe\x64\stream_data_CAN_00.asc

C:\TOOLS\common\MDF\ADTF_2.13.3\exe\x64>
```

1V Tool Chain

➢ Trigger mode setting for ADTF recording

If u want set trigger mode for ADTF2.13.3, u can change XML in MEA folder, detailed info like this, already verified on Chery 1V project：

<Options>
<SplitAfter>60</SplitAfter>
<StopOnDiskSpaceUsage>90</StopOnDiskSpaceUsage>
*<PreTriggerDuration>**X**</PreTriggerDuration>*
*<PostTriggerDuration>**Y**</PostTriggerDuration>*
</Options>

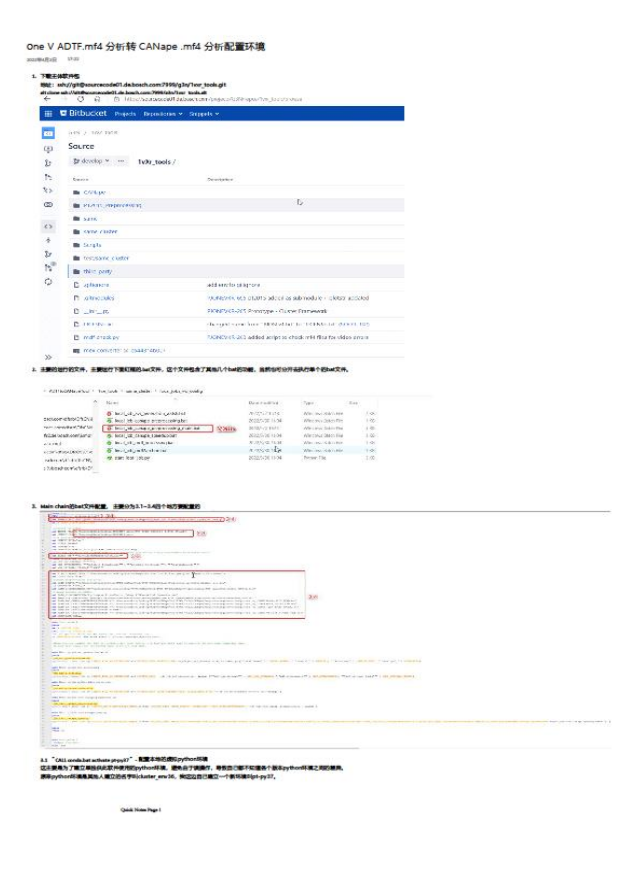
U can adapt the time u want, if u press ‘A’ the recording starts with the time u set before and after, recording should stop automatically.

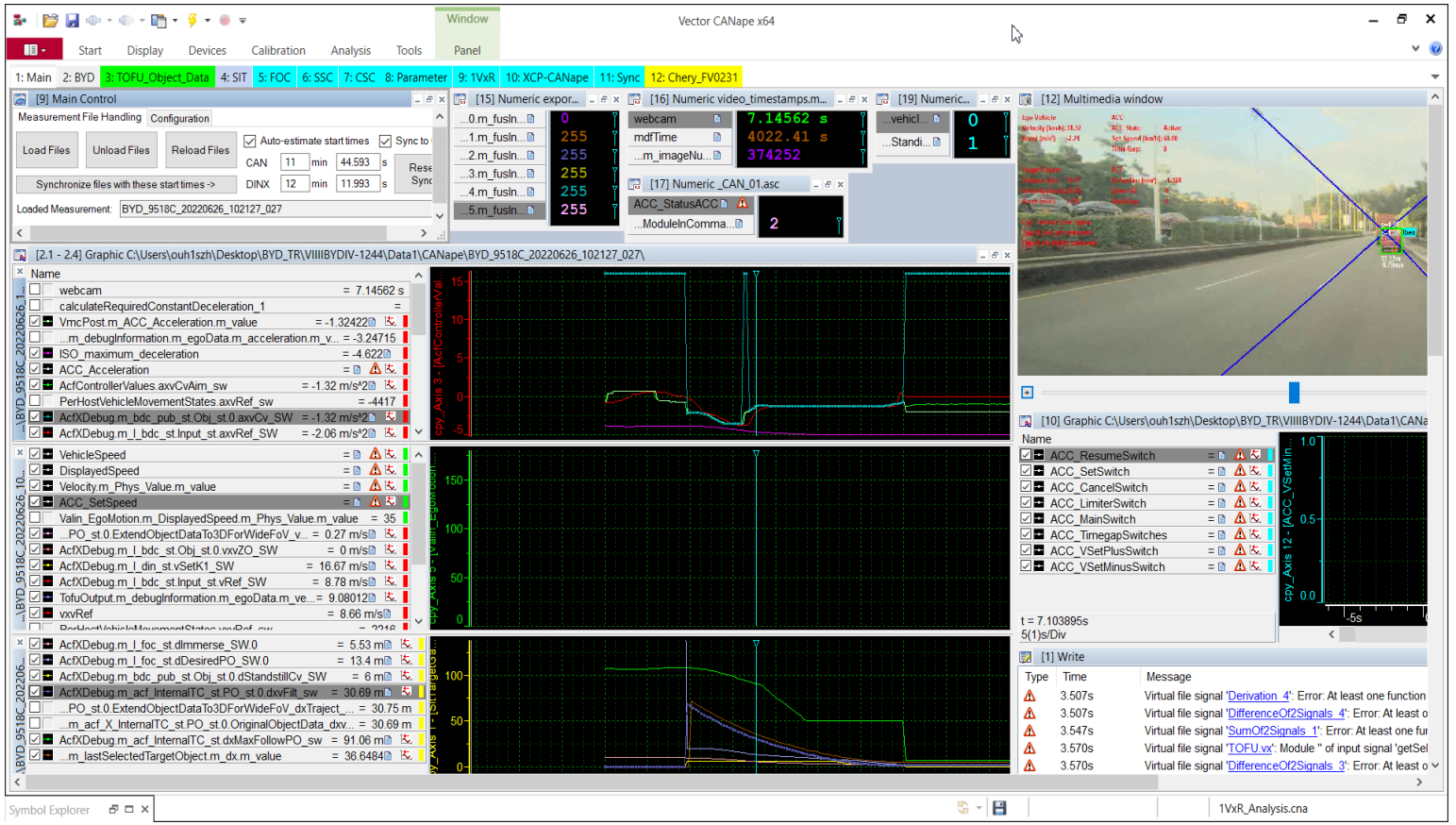
Note:

1.For ER vehicles, 5 is highly recommended for Diskspaceusage(rbcopy is mandatory for ER vehicles). For APP vehicle, 90 is good.
2.If ADTF data need to run **HIL** simulation or **LESS**, **VVS check** is mandatory, especially for **CV simulation**.

1V Tool Chain

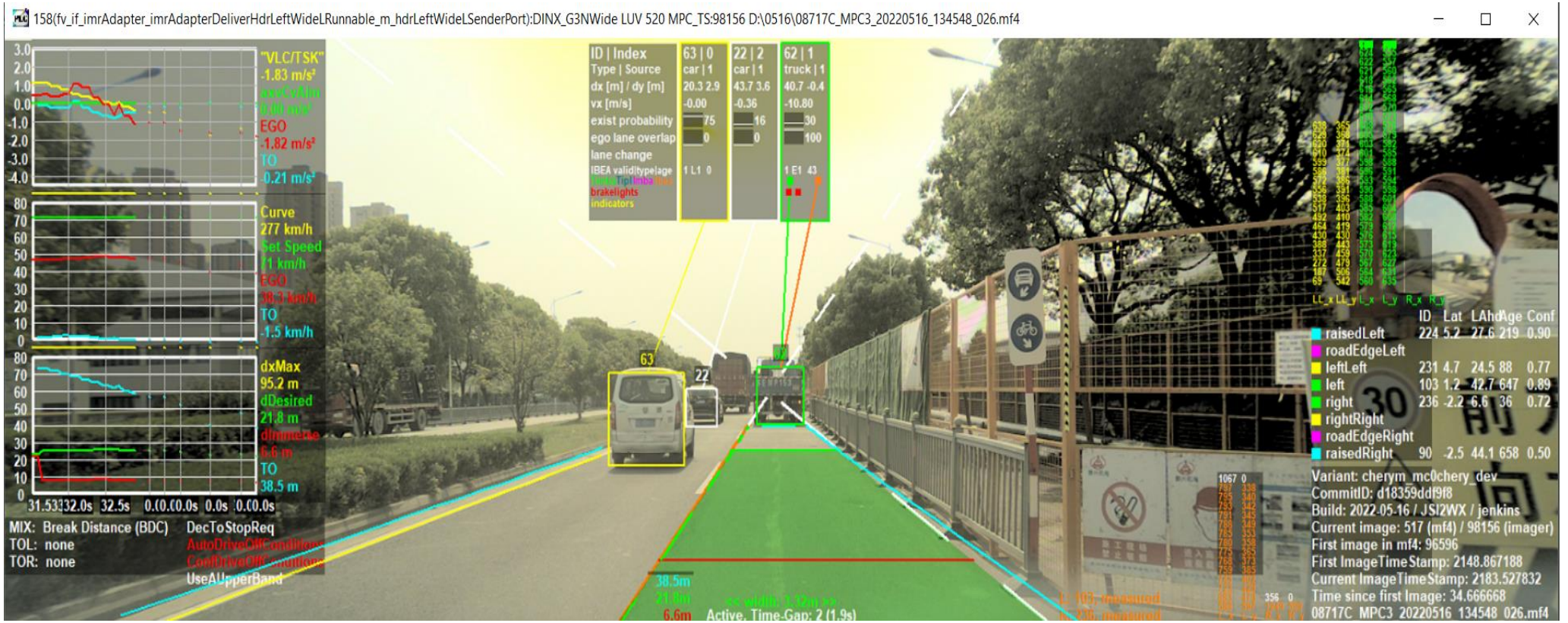
➢ ADTF MF4 change to CANape MDF





1V Tool Chain

- Brief data analyze based on Watch



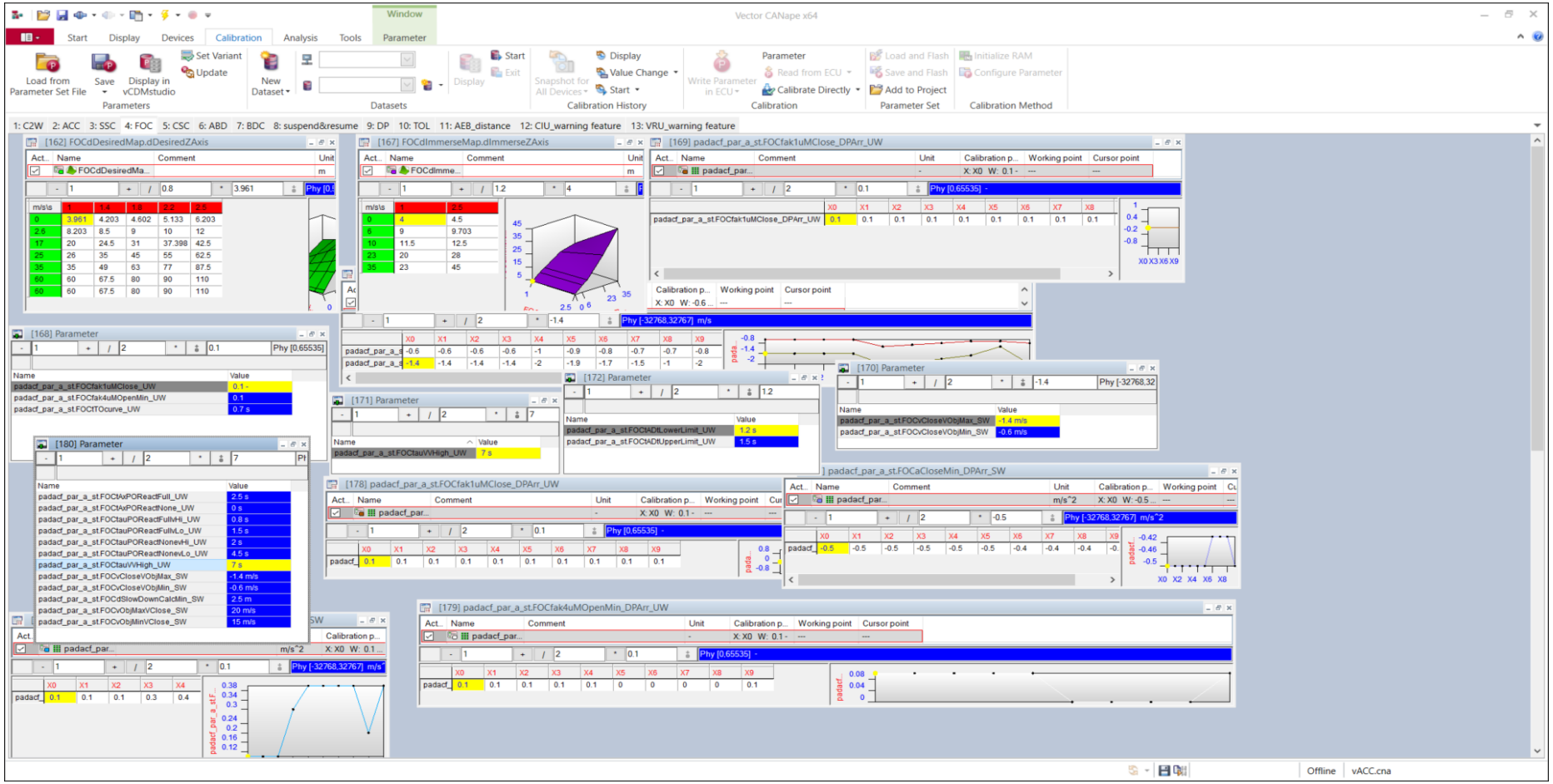
1V Tool Chain

➢ Brief data analyze based on Watch

EX NV FN EP and SA are conditions which need to be met in order to issue an emergency brake.
EX = Is execution time for an aeb reached?
NV = Is the necessity valid?
FN = Is the feature specific necessity behavior valid?
EP = Is existence probability high enough?
SA = Is self assessment high enough?

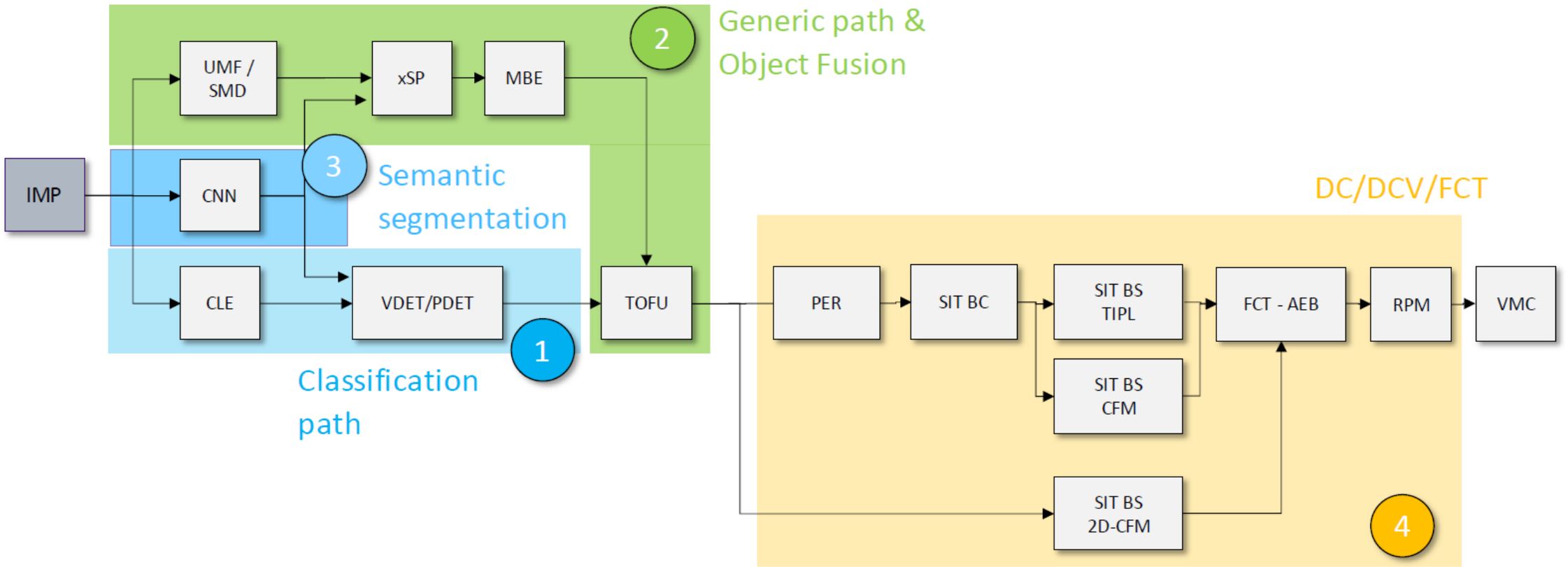
1V Tool Chain

- Online Calibration based on XCP on CANFD



AEB Function Chain

AEB Function Chain ：



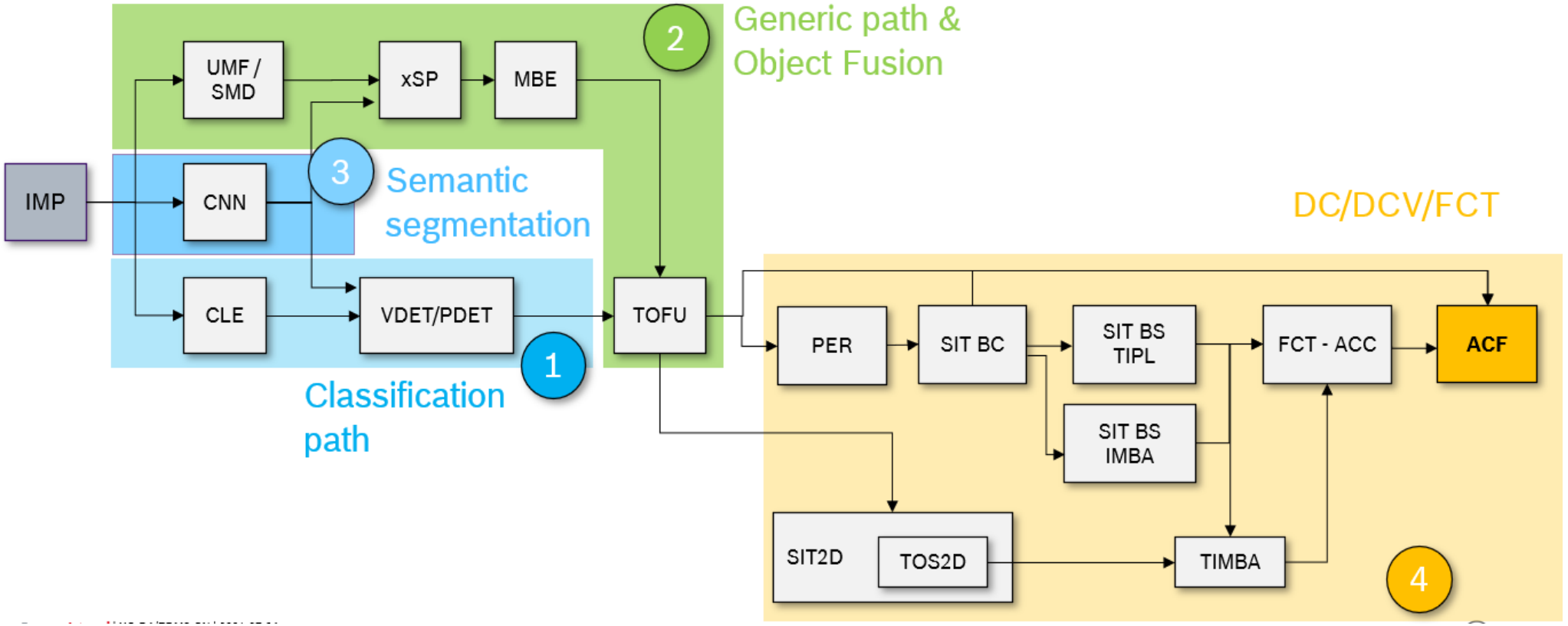
vAEB ER Simulation Result

Chery Brazil ER Summary(200hs)	
Simulation hours	164.8
Event	Activation/FP
AEB	5/0
FCW	71/2
EBA	1/0

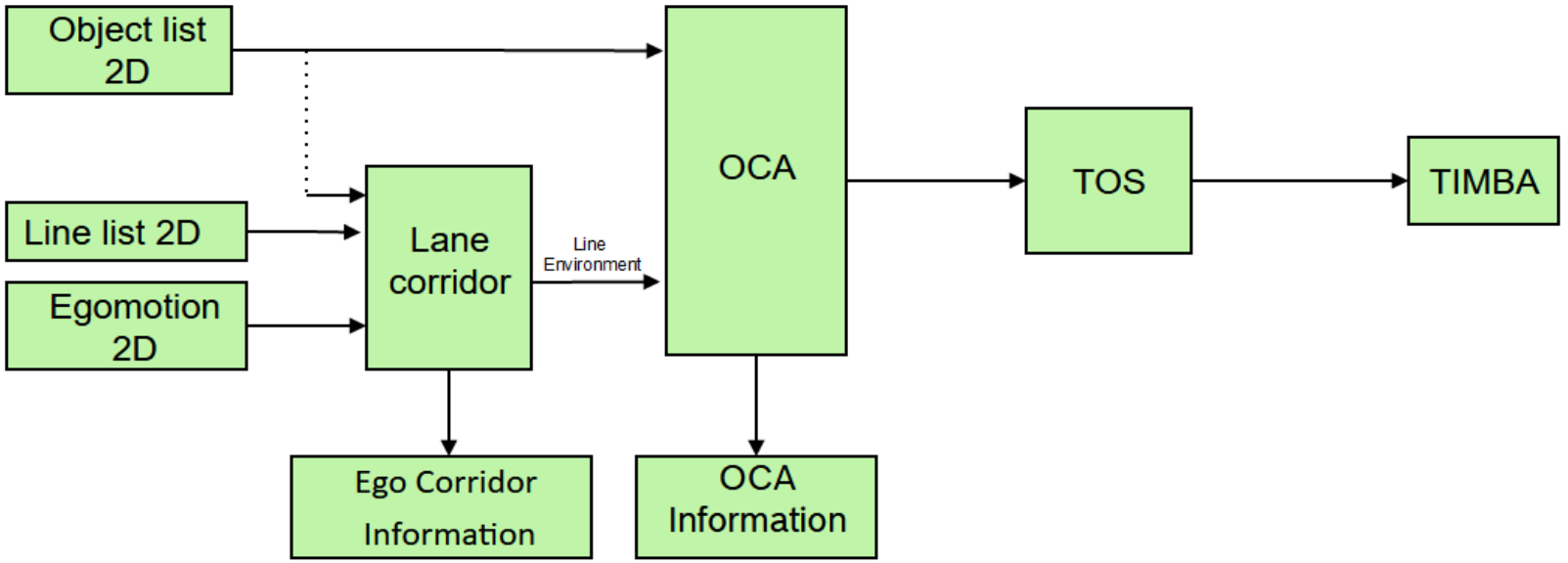
Chery CN ER Summary(1000hs)	
Simulation hours	458.7
Event	Activation/FP
AEB	1/0
FCW	142/7
EBA	1/0

ACC Function Chain

ACC Function Chain：

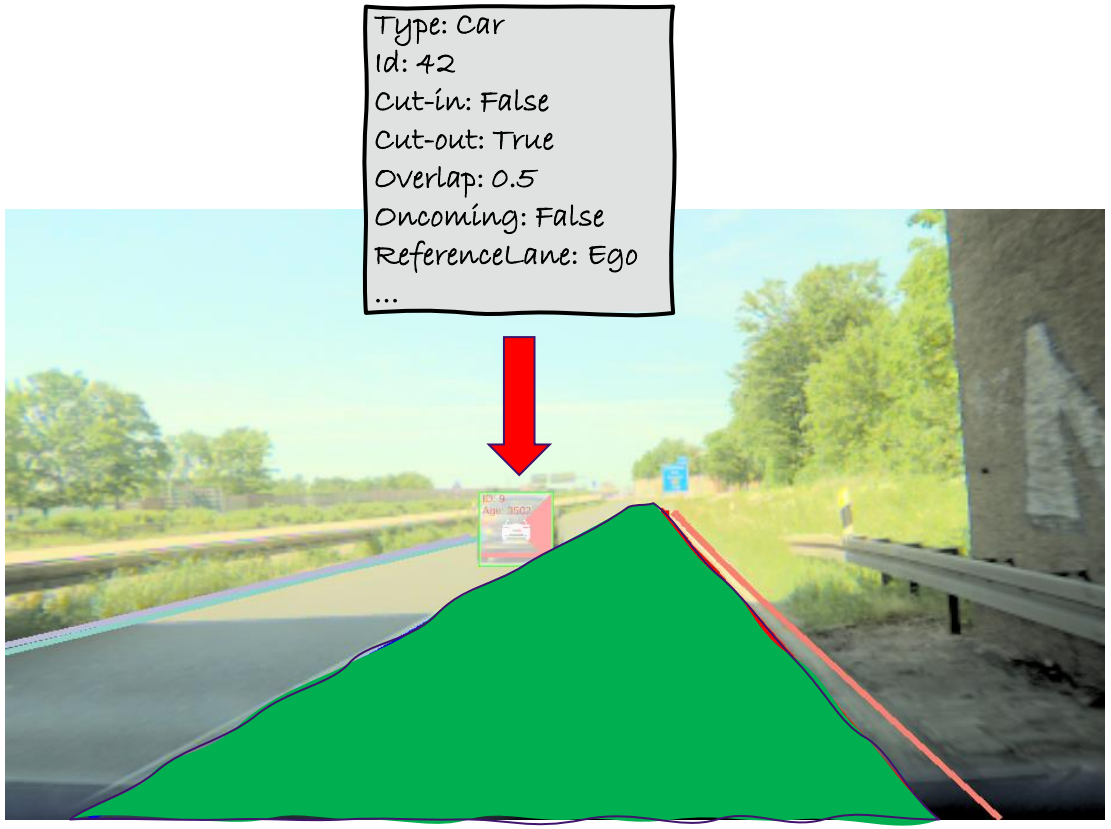


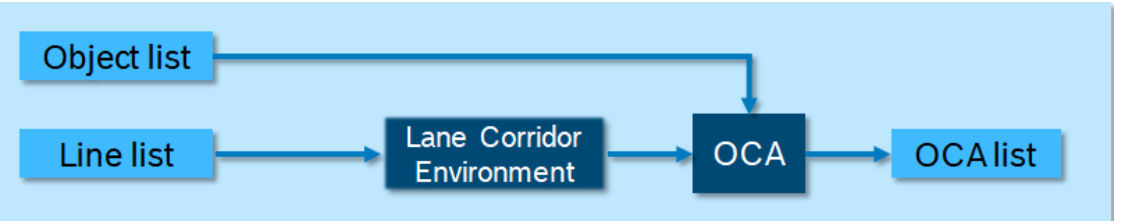
TOS2D Architecture



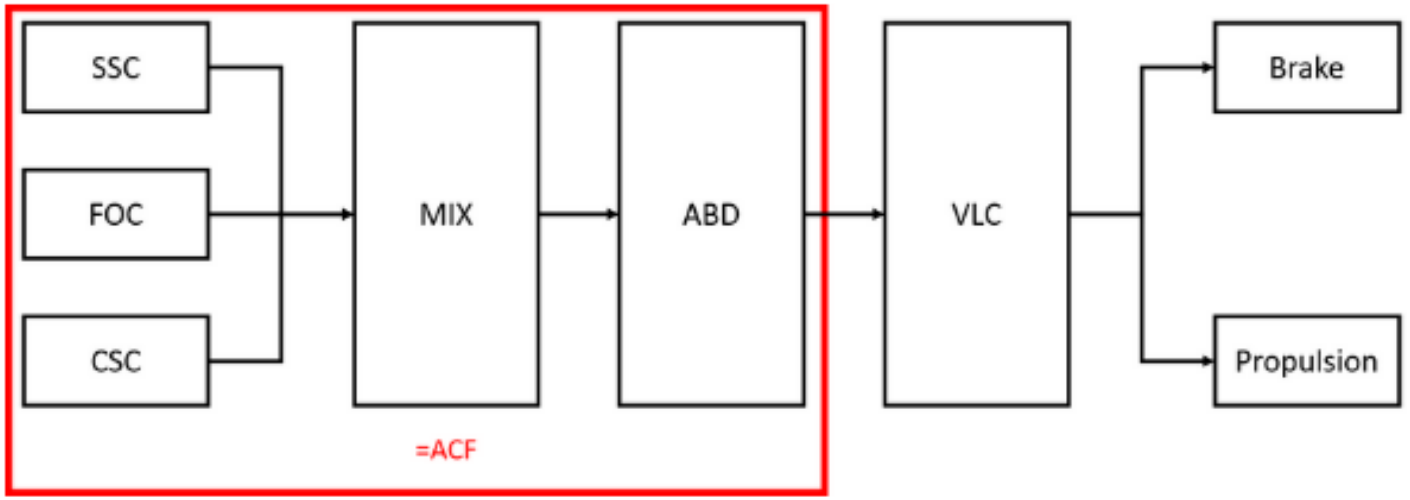
2D OCA – Object-Corridor Association

- 2D Object Corridor Association (OCA) describes the relation of an object to a reference lane
- Based on image based 2D measurements
- Provides additional information
  - Oncoming
  - Cut-in/Cut-out
  - Overlap
  - …
- Input for functional system parts, e.g., ACC, AEB
- Single frame based
- Input: Object list and lane environment
- Output: Object list per reference lane (ego, left, right)



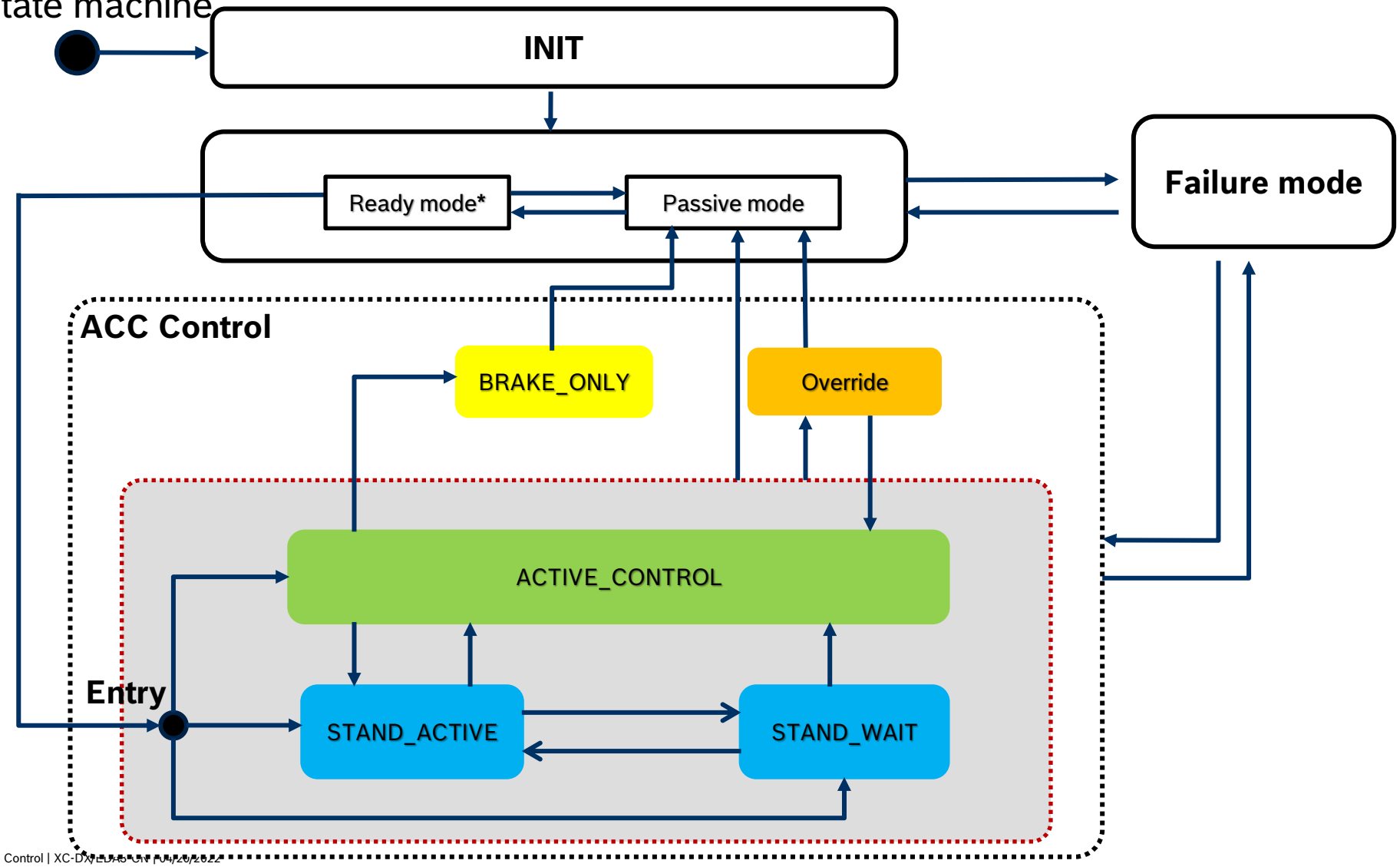


vACC overall logic

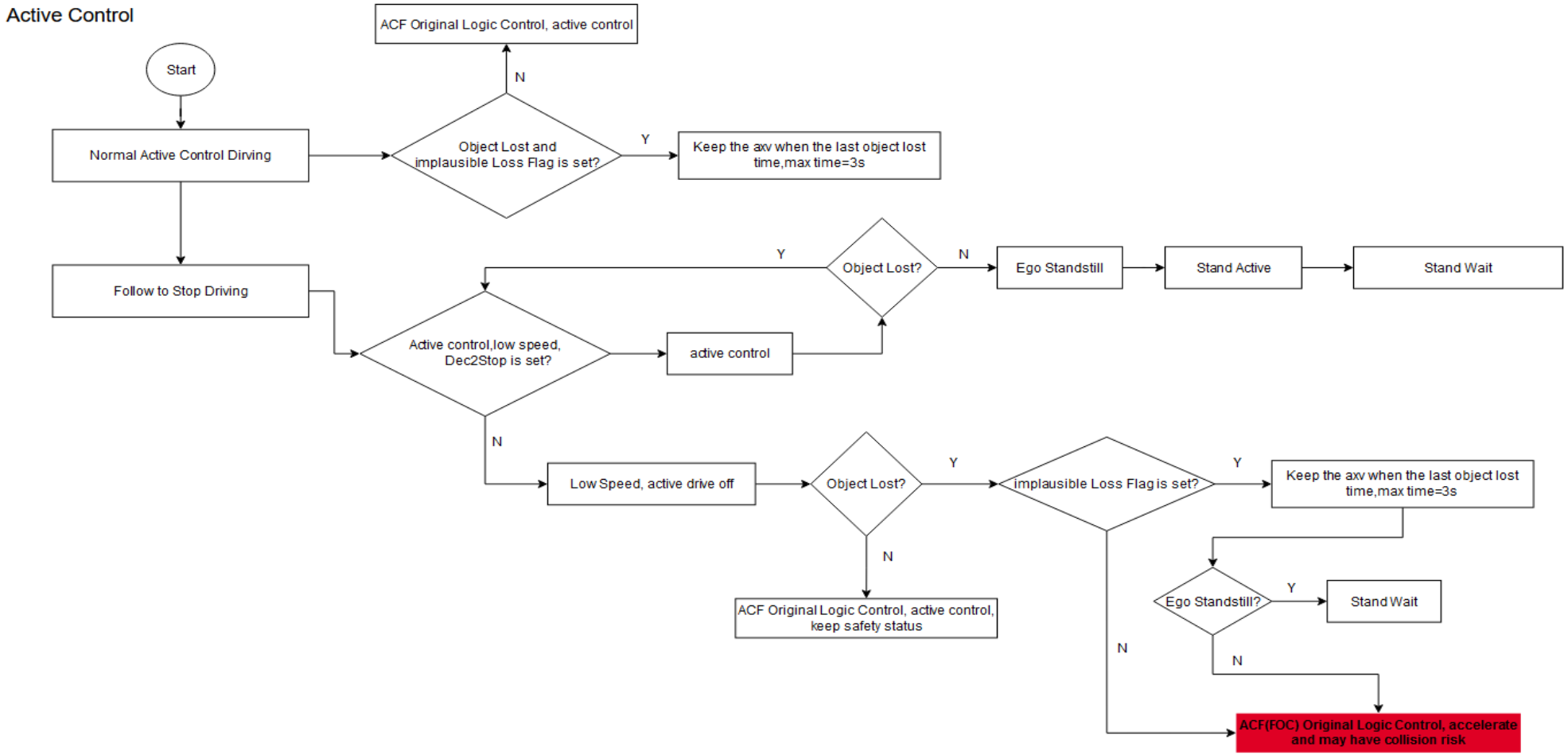


The ACF contains two parts, the platform and the customer code. The customer code supplies specific and additional functionalities to each customer project. Furthermore in the customer code parameters can overwrite the platform parameter values.

vACC state machine



1V TOL logic



Code path:C:\01work\Code\chery_1v_mpc\fvg3_chery\dcv\dc_pj1v\acf_base\modules\foc\foclib.c

vACC KPI

Distance overall (in m)	2498207.0

TOL [/1000km]	
a)	b)
stationary situations	
0	265
265	
ACF	111

total duration:	1286.681221	s

CutIn overall	
Mean	**1.415506419**
Standard deviation	**0.685888544**
Minimum	**0.132675**
Maximum	**3.869519**
Number	**346**

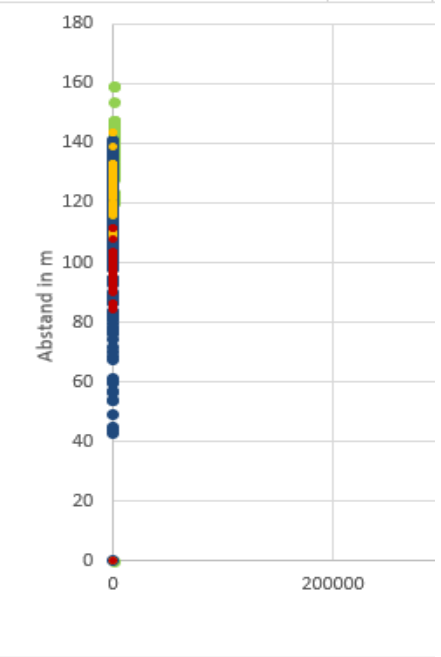
CutIn overlap	
Mean	**0.456691545**
Standard deviation	**0.210585949**
Minimum	**0.025508326**
Maximum	**1**
Number	**346**

NLD [/1000km]		
a)	b)	c)
stationary situations	distance < 30 m	special situations
0	8	21
29		
ACF	-	

total duration:	110.184642	s

CutOut overall	
Mean	**2.421640507**
Standard deviation	**1.170128166**
Minimum	**0.131832**
Maximum	**6.655631**
Number	**540**

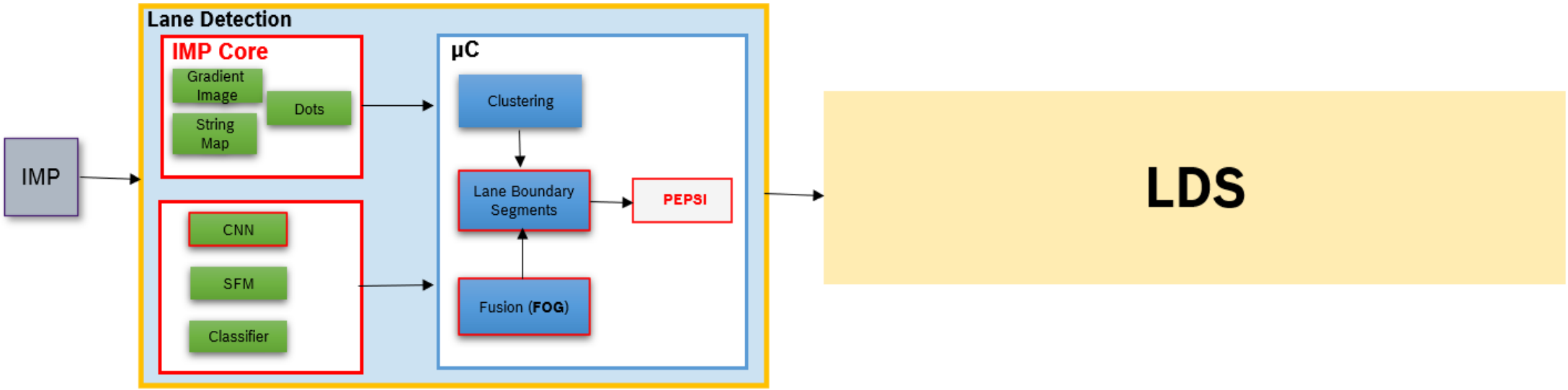
CutOut overlap	
Mean	**0.303792445**
Standard deviation	**0.245166735**
Minimum	**0**
Maximum	**0.969561727**
Number	**540**



Approach	
Mean	**117.22**
Standard deviation	**20.7838**
Minimum	**42.7891**
Maximum	**141.156**
Number	**306**

Lane Function Chain

Lane Function Chain：



Function debug port in ADTF

vAEB debug

- dcv_spp_sppVmcPostprocessing_sppVmcPostprocessing_m_vmcPostSenderPin
- cpj_ov_fct_fdm_ov_fct_fdm_m_portSelectedBehavior
- cpj_ov_fct_fsm_ov_fct_fsm_m_portFctFsmDaFunsStates
- cpj_ov_fct_fsm_ov_fct_fsm_m_portFctFsmDebugOut
- cpj_ov_fct_fsm_ov_fct_fsm_m_portFctFsmInternalOut
- cpj_ov_fct_hmi_ov_fct_hmi_m_portFctHmiDafunctionLane
- cpj_ov_fct_hmi_ov_fct_hmi_m_portFctHmiInternalOut
- cpj_ov_valin_ov_valin_runnable_m_ov_valin_DriverInput_SenderPort
- sit2d_m_SIT2DDebugInfoSenderPin
- sit2d_m_SIT2DOutputSenderPin
- RunnableTiplForwardCollisionAvoidance_m_evaluatedBehaviors_out
- RunnableCfmFrontTrafficCoDriverLon_m_evaluatedBehaviors_out
- PerPmeRunnable_m_pmePort_out

LDS debug port

- lds_lds_runnable_m_ldsCfgCustOutputSenderPort
- lds_lds_runnable_m_ldsCfgPfOutputSenderPort
- lds_lds_runnable_m_ldsBaseDbgOutputSenderPort
- lds_lds_runnable_m_ldsBaseFctOutputSenderPort

vACC debug port

- AcfControllerValues
- AcfXDebug
- AcfXGeneral
- VMCPsostProcessing

1V Tools releated

- ADTF User guideline
- CAN data extract from MF4
- Trigger mode setting for ADTF recording
- vAEB sequence tagging tips
- ADTF MF4 change to MDF 

Reference tip: https://inside-docupedia.bosch.com/confluence/display/EDACN/07+1V+Tools+releated

BOSCH

Q&A