Concept paper of Master Thesis -Begin 01.04.2019

***Research on the embedded board requirements for the real-time evaluation of ADAS functionalities***

**1.Introduction:**

Advanced Driver Assistance Systems (ADAS) applications which is focused on safety applications of vehicles that protects drivers and reduce accidents is expected to increase in the next decade. Although ADAS are still in their early days, most of the vehicle manufacturers realized that it could be eventually the main features in near future since customers are showing more interests on ADAS applications that promotes safety and comfort. Many semiconductor manufacturing companies e.g. NXP, Texas Instruments come forward to develop ADAS application products. NXP has introduced SBC-S32V234 evaluation board which can be used in some important ADAS applications.

In my master’s thesis I will do research on SBC-S32V234 nxp embedded board to evaluate its capability on LKAS applications. At first the board will be evaluated in Hardware In the Loop (HIL) application. Later the board will be evaluated for Vehicle In the Loop (VIL) application. The topics which will be discussed throughout this concept paper of master’s thesis are: Goal of the project, Project structure, Overview of software and Hardware, Project plan, lessons learned and work progress.

**2.Goal:**

The project goal is to research on the ADAS/AD HW requirements and evaluate embedded board capabilities.The board will be evaluated with respect to the following parameters:

1. Performance of the interface.
2. Number of processors required and its performance.
3. Flexibility to the evaluation software.
4. Tools to develop the application.
5. Power consumption.
6. Temperature dissipation.
7. Memory requirement.

**3.Project Structure:**

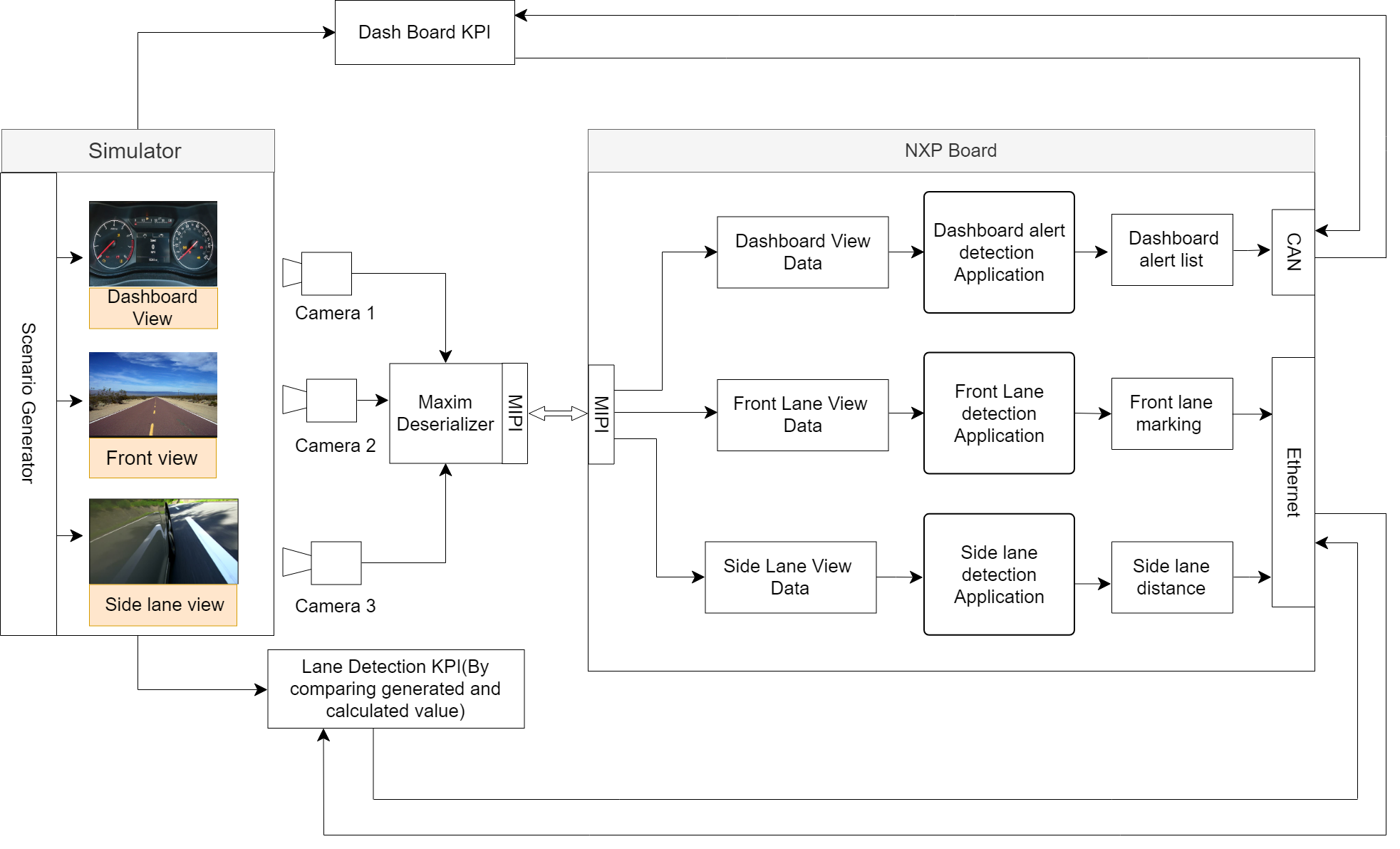
****

Figure1: Project Structure

The setup consists of simulator and embedded board. At first the required scenario is generated using simulator. Total eight cameras can be connected to the board. In our project we need at least three cameras which will be used to provide front view, dashboard view and side lane view. The fusion of different camera data will assist in the Lane keeping assistance evaluation. The camera data is transferred to the board via maxim deserializer. After that the required image processing is done on the APEX core of the embedded board. The calculated parameters will be sent to the simulator via CAN Bus or Ethernet.

**4.Project Milestone:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Setup a Lane keeping assistance system evaluation using NXP SBC-S32V234 Evaluation Board and Simulator | | | | | | | | | | | | | | | | | | | | | | | | | | | | | Date:02.08.19 | | | | | | | | |
| Year | 2019 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Month | April | | | | May | | | | June | | | | July | | | | Aug | | | | Sep | | | | Oct | | | | | Nov | | | | Dec | | | |
| Week | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| S32V234 Board Evaluation |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |
| S32DS Software Installation |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |
| Running Demo project on Emulator |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |
| Preparation of SD Card |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |
| Creation of own Demo Project |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |
| Camera setup |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |
| Analyzing Vision SDK architecture and change in required programming |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |
| Receive images stream from the simulator along with its ground truth data |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |
| Image processing and calculation of required parameters on NXP Board |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |
| Sending Image and required parameters to Simulator via CAN bus and Ethernet Bus |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |
| Comparison of calculated data with ground truth Simulated Environment |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |
| Probable errors fine tuning |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |
| Documentation and writing thesis report |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |

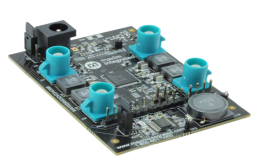
The details overview of the project milestone is as follows:

1. At first the NXP hardware will be evaluated for the autonomous feature’s validation, e.g. LKAS.
2. The scenario on the simulator will be created and stream the video and environment parameters e.g. CARMAKER, VTD.
3. Camera setup in front of monitor with necessary covering to reduce the noise during processing.
4. Required Hardware setup (cameras will be focused to 3 simulators display which will display the front view of lane, dashboard view and right side view of the lane.
5. At least 3 cameras should be connected with the Board to provide 3 different video data which are dashboard view, left side view and right-side view.
6. The board should provide one CAN Bus data, one Ethernet data and at least 3 camera inputs.
7. The camera images will be grabbed into the board via 3 cameras.
8. Image processing and calculation of required parameters will be done on NXP Boards.
9. The processed image and required parameters will be sent to simulator via CAN Bus or Ethernet Bus.
10. The calculated data will be compared with ground truth simulated environment.
11. Feedback is communicated to the processing unit for fine tuning the processing algorithm.

**5.Cost Analysis(approx.)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | Cost in € | | | | |
| Quantity | Work packages | Preparation | Measurement Setup | Execution | Report | Total (€) |
| 1 | NXP SBC-S32V234 Board |  |  |  |  | **626.290** |
| 8 | MXOV10635-S32V Camera |  |  |  |  | **1158** |
| 3 | MAXIM deserializer |  |  |  |  | **106.150** |
| 4 | Monitors(3) |  |  |  |  | **450**Approx |

**6.Overview of the NXP Board, Software and other components [1]:**

Simulator Monitor OV10635 Camera Deserializer



NXP-SBC S32V234

Figure 2:NXP Board and other components

The *SBC-S32V234* is a low-cost development platform for the *S32V2* vision processor. The Module board contains *S32V234* processor, memories and power regulators for the module board and the Carrier board contains all system specific I/O like camera connectors, Ethernet port, display port, SD card slot, CAN ports etc. Total 8 cameras can be connected to the Board through deserializer. In this project *MXOV10635-S32V* camera have been used. The deserializer board model is *MAX9286S32V234. S32DS 2018.R1* for Vision (Contains eclipse neon 4.6 framework) IDE from NXP will be used for simulation.

**7.Lesson learned and Work progress:**

At first *S32DS 2018.R1* software were installed on Ubuntu *18.4* version. But *S32 DS* for Vision only supports Ubuntu *16.4*version and openjdk-8. After installing Ubuntu *16.4* version and openjdk-8 the required software has been installed successfully. From APEX graph project, APEX program project and APEX application project is created accordingly using S32DS software on Linux. ISP dataflow project and ISP application project is also created using *S32DS* for vision software. The VSDK architecture is analyzed. The new APEX2 kernel project is created. The up sampling and down sampling of image is done using own developed kernel and prebuilt kernel. The RGB image converted to Grayscale image using APEX core framework. The frame output buffer configuration is also done for display the larger image.

**Hardware Setup:**

1. On S32-SBC PWA jumper connected to 12V power to power deserializer board.

2. On S32-SBC the MAX deserializer board is connected to the MIPI-A.

3. On MAX deserializer board jumper JU4 is set to power the camera from SBC board (After setting the jumper the yellow light of each cameras are lit).

The full hardware setup image is as follows:



Figure2: Hardware Setup

**Software Setup:**

The SD card is prepared with *SDK\_S32V2\_RTM\_1\_3\_0\_img\_yocto.tar.gz*.After preparing the SD-card, it is inserted in to the evaluation board. Camera captured images will be displayed on the HDMI connected display unit. Power on the boot, login and change directory to the demo folder. Then the sample application *isp\_ov10635\_quad.elf* was running on the board using the below command

*root@s32v234sbc:~/vsdk# ./**isp\_ov10635\_quad.elf*

**Challenges:**

The first challenge was to setup the environment variables on Linux operating system to install *S32 DS* for Vision *2018.R1*. Previously Ubuntu 18 and openjdk-10 was installed on our Linux operating PC. But *S32 DS* for Vision *2018.R1* only supports Ubuntu-16 and openjdk-8. After installing openjdk-8 and ubuntu-16 the software was working perfectly. At first *SDK\_S32V2\_RTM\_1\_0\_0\_img\_yocto.tar.gz* was used to prepare the SD-card. After that, we came to know that the *SBC-S32V234* board only support the BSP file called *SDK\_S32V2\_RTM\_1\_3\_0\_img\_yocto.tar.gz*.To provide 12V power to each camera from the deserealizer we need to set board jumper JU4.Since, there are no proper documents which describes the kernel graph project, it took a little bit more time to learn more about Vision SDK architecture.

**References:**

[1].<https://www.nxp.com/support/developer-resources/evaluation-and-development-boards/ultra-reliable-dev-platforms/s32v-mpus-platforms/s32v-vision-and-sensor-fusion-evaluation-board:SBC-S32V234>