

SAT Feature Planning Lists and Details

y2023

S AT	0 01	Reduced memory footprint for non-IVI use cases	
S AT	0 02	Landscape mode demo apps for CES	
S AT	0 05	Binary Package feed via Yocto or other method	
S AT	0 06	Software Update	
S AT	0 07	LTS UCB (Smart reusing source code of well-maintained packages like Redhat or Debian distributables)	
S AT	0 08	Next-gen In-vehicle cockpit Linux system architecture beyond existing IVI and IC	
S AT	0 12	Adopt camera middleware, contribute missing use-cases and improvements upstream	
S AT	0 13	Camera API use-case: Rear-view camera RVC (backup)	
S AT	0 14	Camera API use-case: Stitched 360-degree all-around vehicle view	
S AT	0 15	Camera API use-case: Front-facing camera (incident reporting, ...)	
S AT	0 16	Camera API use-case: Driver-facing camera (attention detection, ...)	
S AT	0 19	Update AGL System Spec. Latest version was completed in 2015	
S AT	0 1 9.1	Create Production Requirement SPEC:Toyota will create a production requirements document based on the Basesystem which already contributed. As the next step, we believe that it's necessary for each OEM to extract requirements from this document as a reference.	
S AT	0 20	RISC-V support for UCB	
S AT	0 2 0.1	RISC-V QEMU with graphics	
S AT	0 2 0.2	RISC-V board	
S AT	0 21	Backport revised Yocto CVE checker to Dunfell (Lamprey LTS) and Kirkstone (Octopus and beyond)	
S AT	0 22	New demo hardware set to replace Green Machines	
S AT	0 23	Fully UI design upgrade to reach feature vehicle. Target IVI, IC EG Cluster, other? Candidate : Based on Suzuki design that contributed by Haraki-san in last year. IC-EG has UI upgrade plan to harmonize to this design (that task is inside a EG plan).	Existing AGL Qt IVI design was not change from y2017. The flutter based UI is no big improvement from Qt IVI design. This mean, AGL UI design has not update around 6 years. I think so, AGL must re activate in this year. This 1st step is update UI design to show AGL activity to all of world. Haraki-san share the another base design for AGL UI. Shall discuss based on that design. This topic is not required to green box update.
S AT	0 24	Qt update to 6. Need or not need for UI upgrade.	Qt5.15 is ended maintain. Shall migrate to Qt6.2 LTS or Qt6.5 LTS? When AGL will update UI design, IC and Qt IVI shall migrate to Qt6.x. This point is depend on SAT-023.

S AT	0 25	New board support. Reach out to AGL member company. Existing: Renesas, Qualcomm, Intel, NXP, TI, SanCloud. New: Mediateck, NVIDIA, SiFive, Telechips,	Many SoC bender is already join to AGL. Renesas, Qualcomm, ARC, Intel, Microchip, NVIDIA, NXP, TI(San Croud), SiFive, Telechips. SAT shall reach out to non existing AGL support SoC bender. Microchips, SiFive, Telechips, and more. When these bender have a motivate to AGL official SoC/board, SAT recommend to that SoC to SC/AB. AGL must require to SoC bender support without NDA. Good example is Renesas.
S AT	0 26	Qualcomm Support - work at AB level to get meta-qcom to support 6155 and 8155 BSPs rather than supporting individual companies with different BSPs as well as a plan for future chips.	
S AT	0 27	Benchmarking different boards for availability and usability as well as performance.	
S AT	0 28	Create AGL architecture diagram for each profile. to make it easy to understand which package is needed and why.	Create architecture diagram for AGL each profile. AGL IVI (Qt, flutter, HTML5) IC and IC container. V2C SDV others. Ex. IC container. Function Block Assignment Safety monitoring and real time function which includes device access shall be assigned outside of AGL. - All of the other cluster function shall be assigned onto the cluster container. The diagram illustrates the functional architecture across four layers: Container, Container host, Linux Kernel, and SoC/MCU. Other Container: Empty. IVI Container: Contains blocks for Radio, Audio, HFP, Tuner Control, Turn by Turn, Traffic Sign, CarPlay, Android Auto, Sound Manager, Display Manager, Video Stream, SDL, BT, WiFi, USB, Touch Display, and Rear Camera. Cluster Container: Contains blocks for SPEED, TACHO, TEMP, Shift Position, Telltale, Warning, Fuel Consumption, ECO, ACC, Energy Flow, User Customize, Maintenance, Traffic Sign, and etc. Safety / Real Time Function: Contains blocks for FUEL, Alarm, Backlight, ODO/Trip, DIAG, DTC, System, CAN-NM, EOL, LIN, CAN, etc., and Safety Monitor. Container host: Contains Input Manager and DRM Lease Manager. Linux Kernel: Contains ICCOM and DRM. SoC: The main processing unit. MCU: Connected to the SoC. RTOS / Non-OS: Layer between Linux Kernel and MCU. Data Flow – 3. Window Manager Multiple container DRM sharing shall be done by introducing DRM Lease Manager. - GPU rendering/composition shall be done in application container, not container host. - It allows application container to render directly to the DRM device. - It ensures other containers can still display their HMI via Weston. - It allows both types of containers to render to the DRM device in parallel. The diagram shows the data flow for the window manager across the same hardware layers as the previous diagram. Other Container: Empty. IVI Container: Shows IVI-UI rendering to a Wayland Compositor, which then uses libdrm to interface with the DRM device. It also includes an IPC connection to the Cluster Container. Cluster Container: Shows Cluster-UI rendering to a Wayland Compositor (which also receives input from the IVI container via IPC) and a Direct Rendering path to libdrm. Both paths then interface with the DRM device. Container host: Contains the Lease Manager, which coordinates access to the DRM device between the containers. Linux Kernel: Contains the DRM driver and Hardware Composite block. SoC: Features two output paths, CRT1 and CRT2, leading to the Center Information Display and Cluster TFT Display respectively. MCU: Connected to the SoC. RTOS / Non-OS: Layer between Linux Kernel and MCU.