



Demo implementation of ELISA safety architecture for Automotive Grade Linux

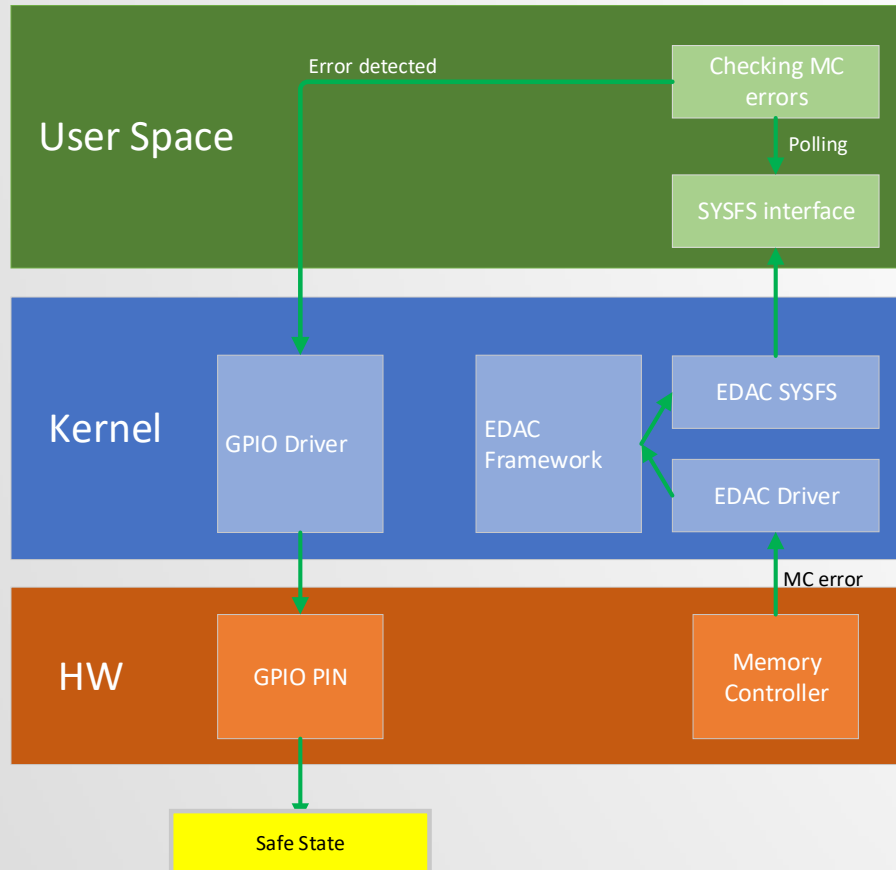
AGL/ELISA collaboration team

Overview

- ELISA safety architecture working group discuss and develop to EDAC based architecture.
- AGL/ELISA collaboration member think to implement this example on AGL.
- This slide describe this collaboration detail.

Current ELISA example architecture

Memory error use case



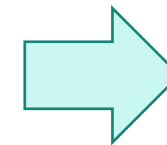
- System definition
 - Memory error is detected by Memory Controller (HW).
 - EDAC driver passes error information to EDAC framework.
 - These H/W and driver are qualified by SoC bender.
 - EDAC framework and sysfs and safety application are qualified by ELISA community.
 - When Memory Controller detect memory error, safety app asserts the GPIO port to change a safe state.
- In demo environment
 - H/W and EDAC driver reprice to virtual Memory Controller edac device (edac inject) driver to easier to analyze the safety part such as EDAC framework, sysfs and safety app.

Proposal

- Current ELISA example software port to AGL distribution
 - Task 1
 - ELISA will create yocto layer (e.g. meta-elisa) at ELISA github repository.
 - AGL is constructed by yocto build system.
 - ELISA yocto layer should be including d edac inject driver and safety app.
 - Task 2
 - AGL import ELISA yocto layer to distribution.
 - Will try to marge to JJ release (jul. at 2020).
- Both task will do by AGL/ELISA collaboration members.



+



AGL with ELISA demo is running on AGL ref. board



Issue and feedbacks form AGL side review

- Issue
 - Need to edac inject kernel module
 - AGL distribution is running on many boards such as R-Car H3/M3, qemux64 and raspberry pi.
 - These board don't use same kernel version, difficult to backport to each kernel.
 - In this case, should use kernel module (out-of-tree driver).
 - Need to support cross build
 - Yocto build system is cross building system
 - Current dummy app is not supporting cross building
 - Please refer to this page
 - https://github.com/elisa-tech/Safety_Architecture_WG/issues/1

Future

- What we aim
 - Basically we want to use AGL for common demo and evaluate environment.
 - We want to reuse the EDAC analysis result in telldale use case.
 - For example, we will implement display error detection driver using EDAC subsystem.