VisionFive2 EPOS port

1. Introduction

As explained in previous sections, we will use SmartData@EPOS as the backplane of our autonomous car design and implementation. So the team's first effort was to port the Embeddem Parallel Operating System to the chosen Single Board Computer VisionFive2, the SBC is a high-performance RISC-V with an integrated GPU. EPOS already had an implementation for the RISC-V architecture, so the team's job was to develop the SBC UART driver, adapt the memory map for the registers offset and adjust the booting script. These tasks are detailed below. The generated code is in this Gitlab.

2. UART-driver

3. Memory-map

4. Flashing Image

The EPOS image flasher has been integrated into the EPOS library, see Gitlab. The dependencies and step by step for the flash to work correctly are listed in this GitHub README.md

5. Debugging EPOS

To run EPOS, we tested the simple hello test in Gitlab. At first, the EPOS setup was concluded successfully but in the operating system initialization an interrupt was being triggered crashing the initialization process and the test didn't run. After debugging and following the RISC-V documentation, we found that it was a machine external interrupt that was crashing the initialization and the methods meant to disable interrupts weren't doing it for external interrupts. So we updated the disable interrupts method following the RISC-V documentation, writing zero to the third bit(MEI) of the MIE register, disabling machine external interrupts. Now the application is running successfully, printing "Hello World".

6. PLIC for RISCV

Access on link.