Embedded Software Engineer

Software development scenario

You're working on the embedded software for a new satellite bus – a generic platform reconfigurable depending on the scientific instruments that the client aims at integrating. For increased versatility, this bus is composed of various on-board computers, each based on different processors.

Your objective is to design a BSP that fulfils the following constraints:

- Compatibility with various embedded targets (common BSP for all the computers)
- The BSP should support testing on host (Linux PC)

Open subject for reflexion

What constraints would you like to signify for the on-board computer architecture/selection phase? What would be your architectural approach for the BSP to ensure its multi-compatibility? How would this be reflected in your implementation?
How would you sequence/prioritize all the developments?
We selected 2 IMUs for the satellite bus, but the sensors are not available yet. How would you adapt our development and integration phases accordingly?

Technical subject (implementation)

You are in charge of developing the first BSP for an electronic board based on Microchip ATSAMV71Q21B. In order to make some preliminary tests, you need to verify the IOs routing on the board. Your first job is to develop a GPIO driver.

The design and implementation of this driver is up to you, and it can be done in C++ (preferably) or C.

The controls required are:

- Direction/Mode configuration,
- Pull-up/pull-down configuration
- Get IO value
- Set/clear output value

To validate this driver, you discuss it with the electronics engineers in charge of this board. They indicate that the IO configurations must be set as follows:

- **PB12:** input / pull-up
- **PC9:** output
- **PC10:** output
- **PA9:** UART0 Rx
- **PA10:** UART0 Tx
- **PB1:** AFEC 1 Channel 0