Open Questions about Embedded Software

- I chose 3 questions. These are questions 1, 2, and 5.

Q.1. What are the main features of Real-Time Operating Systems and how do they differ from regular OS's?

A.1.

- Have a mechanism for prioritizing between tasks. For example, have a mechanism for applications such as completing a job and having another job started based on the completion of that task.

- Synchronization mechanism to ensure that multiple tasks are performed simultaneously and to maintain data integrity between tasks.

- Delays are very important for RTOS and there is a deadline for tasks.In other words, there is a certain amount of time required for the inputs to be converted into outputs, and it must be completed within this time.

- Normal operating systems differ from RTOS in terms of synchronization and latency. Because in normal operating systems, delays do not matter at all, so they are not used for applications where time is important. In normal operating systems, there is no deadline for the task to be done, and unlike RTOS, the task priority does not depend on which one should occur first.

Q.2. When is it preferable to use STM32 over Arduino, ESP32, PIC or other comparable embedded system?

A.2.

When we look at the features of the STM32, it has advanced libraries, interrupt management is integrated with the kernel, low power mode and is cheaper than other comparable embedded systems and can run at high speeds, so it can be used in applications that require multiple advanced and a series of motions and orientations, industrial devices and for testing a product. We need the STM32 platform when we plan to do it commercially, not in prototype form.

Q.5. Let's say you want about 30 hardware modules to communicate with each other. There is one STM32 microcontroller on each module, and the modules can be removed and installed instantly. Which communication standard would you use for these modules to communicate effectively with each other? Why?

A.5.

If I want about 30 hardware modules to communicate effectively with each other, I use the CAN BUS communication protocol. Because when we look at its features, CAN Bus allows the modules to communicate effectively and reliably with each other without a host computer in the system where it is located. Since we can use up to 70 CAN Buses in the system, we can have 30 hardware modules communicate. It is not specified in the question whether the data will be received as analog or digital, if we use other communication protocols, since the cable length will be important, the errors due to this may increase and it may be not easy to communicate effectively. CAN Bus reduces the number of cables and potential errors with multiplex cabling that combines the transmission of analog and digital signals over a common environment.

It is also advantageous in that it is low cost and there is no need to perform a large-scale transformation of the entire system if we want to add or remove equipment. Other advantages of the CAN BUS protocol are that it can be durable for a long time and is efficient. CAN Bus is resistant to electrical disturbances and electromagnetic interference and provides maintenance-free operation for a long time due to this feature.