Types Of Embedded Systems Programming

One definition of an embedded system is a computer and associated devices that perform a specific function rather than run general purpose programs. It may or may not have a user interface.

The following are one way of looking at embedded systems programming. Of course, there will be overlap between the types. Embedded systems can be microcontrollers on an SBC or large racks of CPU's and device cards. Programmers can have a range of skills that suit them to a particular role.

Bare Boarder

The target is a brand-new bare board fresh from the fab. It’s the first one and is a new design. The job is to bring it up to the point a boot monitor works, and all onboard devices are tested and working at a basic level. The boot monitor can download and run basic single threaded test programs. It isn't known if everything on the board works or there are defects in the design or fab. Developers working at this level may be EE's who program or a team of a hardware only EE and a CS devs. They must be able to read and understand the CPU and device reference manuals. CPU's can be especially complex these days although most CPU vendors will provide a baseline bring up routine that the developer can start with. Tools such as U-Boot can be leveraged. O-scopes and logic analyzers may be used.

Platform Integrator

The target is a known good board that can download and run programs. The job is to provide the application developers with an OS framework that they can use. It may require integrating an RTOS, Linux or another app framework. The target can range from an SBC with all devices onboard, or a bus-based system that requires integrating several modules. Most of the device drivers are available one way or another, from vendors or open source. Developers working at this level could be experienced with OS systems like Free RTOS, VxWorks or Yocto. The devs will start from a pile of known good parts (they hope) and software and integrate them into a platform that the app developers can use.

Application Developer

The target is a known good system with some OS framework. The hardware may come from a vendor as an integrated package. The job is to implement a specified application. At this level the programming is very similar to application programming on a general-purpose computer. A spec and design are developed, and a dedicated application is implemented. The differences may be that it talks to device rather than users (of course there can be a user interface). It might have more definite performance and reliability requirements. It will probably have more constraints than a general purpose program. For example, don't use dynamically allocated memory and have deterministic or bounded execution times. There may be techniques, such as state machines, that are less common in general purpose systems. Developers at this level may not need to know the details of how devices work at the bit level. In that way its like writing programs for a PC.

Domain Expert

This type will be an expert in the specific application domain. Knows everything about it inside and out. They will most likely have years of experience in that specific subject. Their value is in developing the specs and design for the system. They may be a good programmer but are more focused on the system, its environment

and how it has to work. They are typically senior people or it may be a new PhD who knows theory. In both cases they will probably work with application developers to implement the system. An application developer can become a domain expert over time.

Other

Test Engineer

One subspecialty could be a test engineer that is responsible for making sure the system works as advertised. It might include testing the hardware and/or the application.

Devops

Embedded system development these days will require some sort of devops framework. Maybe just git and cmake for a one man show. But teams can be big with dozens of team members. In that case it would be possible to be a devops that specializes in the embedded environment and its unique requirements. In a big team is fantastic to have someone responsible the builds, testing and deployments. And it can be a full time job.