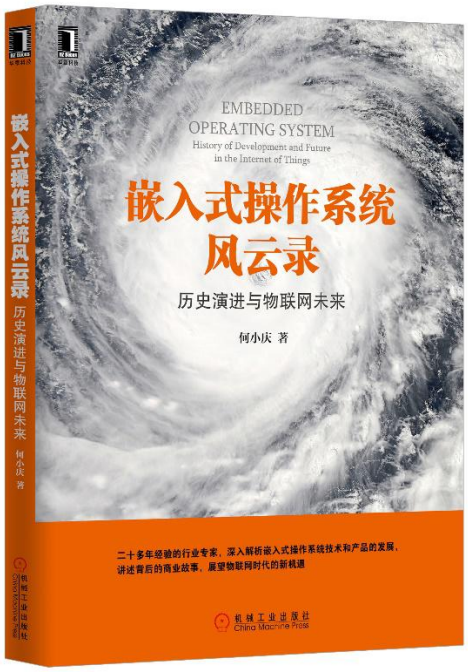
The Choice of Operating System for Internet of Things Device Development

Allan He

Commercial embedded operating system originated from late 70s, and has since experienced 4 stages: RTOS kernel, RTOS, embedded general OS (e.g. embedded Linux and Windows) and application-oriented embedded OS like Android. My authored book " Embedded Operating System: History of Development and the Future of the Internet of Things" has in-depth contents of the subject.



The new generation of embedded OS is IoT OS which is targeting IoT application. There are a couple of factors surrounding the choice made by a development engineer. First off, real-time requirement, which could be hard real-time (e.g. µC/OS-III and VxWorks) or soft real-time (e.g. embedded Linux) depending on the project. Secondly, security concern which include external network threat, and also some applications like high-speed railway and aviation require a much higher security level compared to regular consumer electronics; there are some commercial and open source embedded software solution that address the requirement. Thirdly, open source consideration: open source comes with active community, rich ecosystem and broad support of hardware and standardization which attract many developers. Fouth, variety of tools: with rich third-party tools (incl. development, performance measuring and testing tools) support, development cycle can be shortened. Fifth, cost factor: while open source is free, the actual process of using it is not; commercial software may sometime provide perfect off-the-shelf solution.

According to an IoT developer report in 2016, 70% of current IoT devices are running on Linux, 13% on FreeRTOS and none of the result account for more than 5% (e.g. ARM mbed OS,

Contiki and TinyOS). Personally speaking, there are two routes from IoT device development. One is to base on open source Linux and FreeRTOS for example, which has obvious advantage: IoT Cloud protocol stack, connection IoT protocols and some drivers of IoT chip are included and ready for integration, e.g. WiFi, Bluetooth and ZigBee. This route is by far the most popular one. The other one is to utilize IoT OS. As a one-stop solution, characteristics of IoT system are utilized fully in such OSes. However with less genericity, it is not flexible (consider a company that want to cut an IoT application) and is often larger in size. This route is more suitable for IoT venture projects that develop products fast.

There is another catch for MCU: because of the Processor architecture, only RTOS is suitable. Therefore, developers need to (and should) be familiar with RTOS in general and at least one particular release of RTOS. Some IoT OS kernels are based on RTOS technology, including Huawei Lite OS and MxChip Mico OS. In the future, RTOS will become a fundamental software platform for IoT systems.

Commercial RTOS is more mature in term of technology and application after years of development. There are also many different business model within the industry. In recent years, many complete solution for IoT applications have been produced, for example Synergy platform by ThreadX and Renesas, Zerphery by Windriver and Intel, and Micrium which was acquired by Silicon Lab.

Embedded engineers and makers have distinct preference when developing on IoT devices. Makers tend to prefer open source hardware like Raspberry Pi and Arduino. Many of them are from Internet industry with limited in-depth understanding of embedded system, and they usually like high level programming languages like Python and Javascript and prefer open source OS like Linux and Lite OS as well .

As IoT application gain more complexity, new requirements has begun to arise: increase in number of sensor call for longer battery life, and voice/visual human-computer interaction lead to higher demand for connectivity and display. These are all incentives for new MCU products and embedded OSes that use less power but provide higher performance. In the near future, more and more applications will be based on RTOS technology and newer generation IoT OS. This mean choosing and getting familiar with an IoT-ready embedded OS will be more critical to a product’s success.

Allan He is the founder of BMR, an embedded software company in China since 1995. Allan is an early practitioner in the field of embedded operating systems with more than 30 years of embedded system development and marketing experience in Asia. He was the deputy chief editor of "MCU and Embedded System Applications" journal, and has published more than 60 paper or articles in various international and domestic conferences or domestic journals. He was also the author of book "Embedded Operating System: History of Development and the Future of the Internet of Things".