SysPro Consulting
Where the Software Meets the Hardware

Intel® Boot Firmware for IoT Devices

Open-source alternatives to a UEFI BIOS

December 30, 2023

Jay Talbott
Principal Consulting Engineer
SysPro Consulting

Introduction

Since the invention of the original IBM PC, all Intel®-based computer systems have required on-board boot firmware to initialize the hardware and prepare the system to boot an operating system. Early PCs used legacy Basic Input Output System (BIOS) firmware, which could boot into DOS, Windows 3.1, and other legacy operating systems. It also provided basic I/O services to the operating system and applications. Modern PCs use BIOS firmware that is compliant with the Unified Extensible Firmware Interface (UEFI) specification, which supports booting into modern versions of Microsoft Windows as well as other operating systems and provides a much broader set of runtime services.

While a UEFI BIOS is the modern mainstream firmware solution for desktop PCs, laptops, and servers, it is usually not the best boot firmware solution for embedded IoT devices. The only operating system that really requires UEFI support is Microsoft Windows, which is typically not the operating system of choice for embedded IoT applications. For most IoT devices, the vast majority of a UEFI BIOS's capabilities are not required, nor desired.

Weaknesses of a UEFI BIOS

For IoT devices, a UEFI BIOS has a number of significant weaknesses. A UEFI BIOS is based on a very large and complex code base, the majority of which is not needed for IoT applications. The code structure is difficult to follow and understand, and there are very few people who really know how to work with the code base effectively. Internally a UEFI BIOS is essentially an operating system with a dispatcher that executes various code modules, which are dynamically loaded rather than statically linked. This results in code execution that is not completely deterministic and is often slow to boot. These weaknesses make a UEFI BIOS essentially impossible to certify to any functional safety standards such as DO-178C or ISO-26262.

A UEFI BIOS includes several features that are undesirable for IoT applications. For example, it enables and utilizes System Management Mode (SMM) on Intel® processors, which, when entered, halts all current processing on all of the processor cores for an arbitrary amount of time. If running a real-time operating system (RTOS), a System Management Interrupt (SMI) would preempt any real-time performance of the RTOS, which could have a drastic negative impact on the performance and behavior of the IoT device. An SMI could also have a negative impact if the IoT device supports Time Coordinated Computing (TCC) and/or Time Sensitive Networking (TSN).

A UEFI BIOS also includes a BIOS setup menu that allows a user to modify various low-level system settings, boot devices, etc. While such flexibility may be desirable on a mainstream PC, for an embedded IoT device such settings are fixed for the specific application and not user configurable. For instance, an embedded traffic monitoring system doesn't need a BIOS setup menu.

Another area of concern with using a UEFI BIOS for IoT devices is security. The system firmware runs at the highest level of privilege and needs to be highly scrutinized for any potential vulnerability which could be exploited to compromise the performance or behavior of the system. With a larger and more complex code base the presence of bugs and security vulnerabilities require updates, which are not always practical to install on IoT devices in the field. The UEFI BIOS setup menu itself is a security risk. Allowing changes to the system settings could impact the behavior or performance of the system, including potentially making it completely unbootable.

A UEFI BIOS provides a broad set of UEFI runtime services, which can be used by the operating system and applications to perform functions and obtain information about the state of the system. As mentioned earlier, the only operating system that really uses these UEFI runtime services is Microsoft Windows which is typically not the operating system of choice for most IoT devices. UEFI runtime services are not needed for such IoT applications, and they expose a potential open attack surface that could be used to compromise the system.

A UEFI BIOS typically Includes network boot capability, which means that the firmware implements a full network stack. While network boot capability may be desirable in certain applications, this opens yet another potential attack surface into the system.

Lastly, the code for a UEFI BIOS is closed source, requiring a board or system vendor to buy the firmware from an independent BIOS vendor (IBV) such as AMI, Phoenix, or Insyde (paying royalties for every system shipped), or licensing the source from one of the IBVs and having to build the engineering expertise in-house to customize it for the target application(s).

Enablement of Alternative Boot Firmware Solutions

In the early 2010s, Intel Corporation began developing the Intel® Firmware Support Package (FSP) for various Intel CPU and SoC families. The FSP for each supported Intel platform includes a binary file that encapsulates all of Intel's proprietary initialization code along with the supporting files needed for integration into a bootloader solution. These FSPs enable third parties to develop boot firmware solutions utilizing open-source projects such as coreboot or Intel's Slim Bootloader.

The FSP for each Intel CPU or SoC family contains an Updateable Product Data (UPD) structure which holds parameters for FSP initialization. The boot firmware can read this UPD structure and modify the various parameters for the specific application, providing a vast amount of flexibility and configurability.

There are numerous FSPs available for a wide variety of Intel CPU and SoC families. These FSPs are not just for embedded applications but are also being used for mainstream computing ranging from Chromebooks to hyperscale server systems. The Intel FSP has revolutionized the system firmware landscape and opened the door for alternative boot firmware solutions.

Advantages of Alternative Boot Firmware Solutions

Alternative boot firmware solutions that encapsulate the Intel® FSP have numerous advantages for IoT devices. The bootloader options are often based on smaller and simpler code bases, which are statically linked and have deterministic sequential code execution. They only include what is necessary to initialize the system for the target application. The goal of these boot firmware solutions is to do the bare minimum required before handing off control to the OS:

- Initialize the CPU.
- Initialize the memory.
- Configure the I/O.
- Load and handoff control to the OS.
- Do all of the above quickly and securely.

Due to the simpler design of these alternative boot firmware solutions, it opens the door to the possibility of functional safety certification for IoT applications that require it. While it is still a non-trivial task, it is a more manageable option with these firmware solutions compared to a UEFI BIOS.

With alternative boot firmware solutions, undesirable features such as SMM can be disabled, and there is no BIOS setup menu. The desired system configuration is fixed and specified at build time for the target application. These firmware solutions have no runtime services and do not implement a network stack, significantly reducing the attack surface at the firmware level.

The alternative boot firmware solutions can boot into Linux, an RTOS, a hypervisor, or even a bare metal application – essentially any operating system *other* than Microsoft Windows. For that matter, these firmware solutions have the option to incorporate a UEFI payload that supports booting Windows, but obviously that re-introduces some of the key weaknesses mentioned previously that come with a UEFI BIOS.

Lastly, these firmware solutions are based on open-source projects that are not controlled by the IBVs, eliminating any royalties or licensing fees. Because the code is publicly available, it can be customized as necessary to support any custom, unique, or otherwise out-of-the-box requirements that are common for IoT applications.

Conclusion

Based on all of the above, the following conclusions can be drawn:

- A UEFI BIOS is typically not the best boot firmware solution for embedded IoT devices.
- Intel has enabled alternative boot firmware solutions with their FSPs.
- These alternative boot firmware solutions have numerous advantages over a UEFI BIOS.
- Third parties have been enabled by Intel to develop and provide these alternative boot firmware solutions to Intel customers.

Such alternative boot firmware solutions have already been utilized for a wide variety of IoT applications.

About SysPro Consulting

SysPro Consulting is an Intel® Gold partner that specializes in the development of alternative boot firmware solutions for embedded systems that are based on Intel CPUs and SoCs. SysPro spent several years providing engineering consulting services to Intel, working directly with Intel's FSP team. Since that time SysPro has been developing boot firmware solutions for various Intel IoT customers and has been a subcontractor on numerous U.S. government / DoD programs.

Unlike the IBVs, SysPro is a software engineering consulting company. SysPro does not sell *products* nor charge *royalties*. Instead, SysPro sells software engineering services on a purely time and materials basis. SysPro works more like an extension of their clients' existing engineering teams, offloading the boot firmware development from their clients so that they can focus on their core technology. SysPro can customize each boot firmware solution for the specific project requirements and hardware design for the particular application.

At the end of each project, SysPro provides its clients with all of the bootloader source code and supporting documentation so they can then build and support the boot firmware solution themselves. SysPro's clients can then choose to take it from there, or, more commonly, come back to SysPro for any ongoing support, maintenance, or other changes, or for additional boot firmware development projects. SysPro's value is in its expertise and experience in this field, as the SysPro team consists primarily of former Intel employees and others who have been engaged with Intel in one way or another doing boot firmware development.

For more information about SysPro Consulting, reach out to the author of this paper via the contact information below.

Contact information

Jay Talbott
Principal Consulting Engineer
SysPro Consulting
(480) 704-8045
JayTalbott@sysproconsulting.com
http://www.sysproconsulting.com