

Research Article

Platform Independent Embedded Linux OTA Method

Huseyin Karacali^{1*}, Nevzat Donum^{2*}, Efecan Cebel^{3*}

¹ TTTechAuto Turkey Yazilim A.S, Software Architect, 0000-0002-1433-4285, huseyin.karacali@tttech-auto.com

² TTTechAuto Turkey Yazilim A.S., Software Engineer, 0000-0002-8293-8267, nevzat.donum@tttech-auto.com

³ TTTechAuto Turkey Yazilim A.S., Software Engineer, 0000-0002-2027-0257, efecan.cebel@tttech-auto.com

(First received September 23, 2022 and in final form December 22, 2022)

Reference: Karacali, H., Donum N., Cebel E. Platform Independent Embedded Linux OTA Method. The European Journal of Research and Development,2(4), 243-252.

Abstract

Embedded devices are often used at remote locations. Accordingly, since it is difficult and costly to physically access the devices for firmware updates, the updates are made with Over-the-Air (OTA). OTA is a firmware update method that allows embedded devices to be updated wirelessly without requiring physical access. OTA applications for Embedded Linux devices are platform-dependent. Therefore, the aim of this study is to develop an easy integrated, steady, and robust OTA method. This method runs in the background without any user action on a platform-independent Embedded Linux device and it has a small footprint. The firmware is compressed in this method. The compressed firmware is uploaded to a storage area such as a remote drive, cloud, and repository. The storage area is used efficiently thanks to compressed firmware. The OTA application checks versions of firmware in the storage area and compares them with firmware running on the device. It decides which version is more up-to-date. If the version of the firmware in the storage area is more up-to-date, the OTA application runs in the background and downloads the updated firmware without interfering with the working functions on the device. After the download is completed, compressed firmware is extracted and flashed into the specified memory location. When the device is rebooted after the procedure is completed, the device is going to boot with the new version of the firmware. Consequently, the OTA application can be integrated easily on most Embedded Linux platforms. Besides the OTA application can be enhanced with security and power-safe features.

Keywords: OTA, Embedded Linux, Update, Firmware

1. Introduction

Over-the-air (OTA) is replacing software or firmware to quickly find solutions to software or security problems encountered and to present a new feature to the user on the microcontroller with the newer one. Frequently, the released software from the host machine is uploaded to a cloud or a remote server. After that, that software on the cloud is then downloaded and supersedes the previous one. Thus, the embedded device is updated without any physical interaction.

An over-the-air software update in embedded systems without physical contact has become a huge need in the industry because of the large number of products and the cost of accessing them one by one. This need has become even greater with the growth of the market share of IoT devices. (Brown, 2018).

Moreover, OTA is a desired feature for many automotive software projects with Embedded Linux. However, OTA methods that are open-source or licensed are readily integrated and used in these projects but generally, these Embedded Linux OTA methods are platform-dependent. Therefore, these methods cannot be used for different platforms or require too much configuration to use them. Thus, as the method will explain in this article, platform-independent OTA will provide great ease of use.

2. Materials

Platform Independent Embedded Linux Ota(PIELO) consists of Embedded Linux device firmware development, OTA application development, and, Cloud Drive Configuration sections. At these stages, software development and testing activities were carried out.

2.1. Embedded Linux

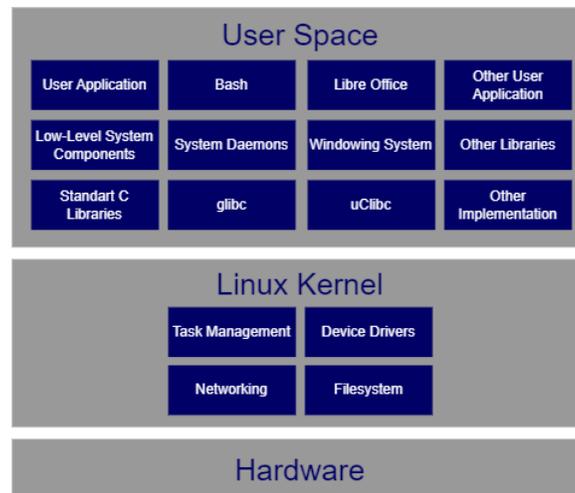
Embedded Linux is a type of Linux-based operating system specifically designed for use on embedded devices and systems. It offers a lightweight, compact, and secure platform that is well suited for these applications. (Barbieri, 2021).

System, Kernel, and distribution viewpoints are taken into consideration when examining Embedded Linux. (Barbieri, 2021).

An embedded Linux system is an embedded system that runs on the Linux kernel. The Linux kernel is a powerful and flexible operating system that is free to download and use. It is highly modular and can be configured to work with a wide range of devices and applications. (Wikipedia Contributors, 2019). For his i386-based personal computer, Linus Torvalds created the Linux Kernel in 1991 at the University of Helsinki. (RedHat, 2019).

Software bundles, external apps, services, and a development framework built on top of the operating system itself are all included in the phrase "distribution." (Barbieri, 2021).

In our project, Embedded Linux was used as the operating system, as it can meet the requirements of hardware support, networking operations, modularity, commercial support, security and, extremely customizable. Simplified Embedded Linux architecture is shown in Figure 1 below.



Simplified Linux System Architecture

Figure 1: Simplified Linux System Architecture

2.2.Yocto Project

The Yocto project was developed by the Linux Foundation by embedded product developers, regardless of the product's hardware architecture. It is an open source collaborative project that helps build, configure, and deploy custom embedded Linux-based systems. (Linux Foundation, The Yocto Project). The Yocto Project is not an Embedded Linux distribution. It helps to create an Embedded Linux distribution according to the developer's requirements and wishes. The open-source project started in 2010 is a project led by Richard Purdie. (Linux Foundation, The Yocto Project). The working structure of Yocto Project is shown in Figure 2 below.

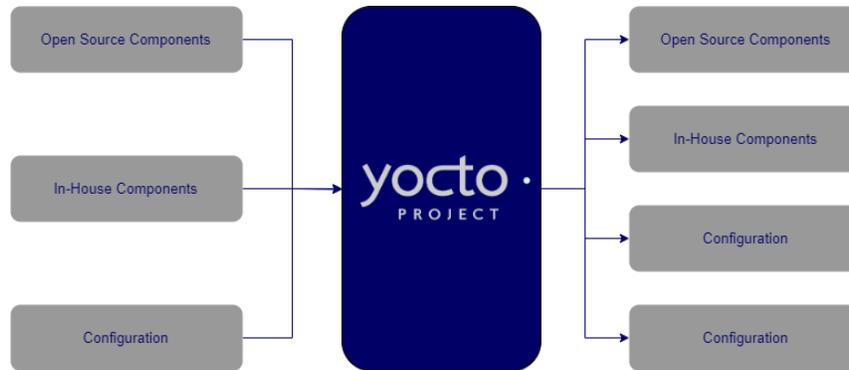


Figure 2:Yocto Project Working Diagram

The three components of the Yocto Projects are:

- OpenEmbedded
- Poky
- Yocto Project

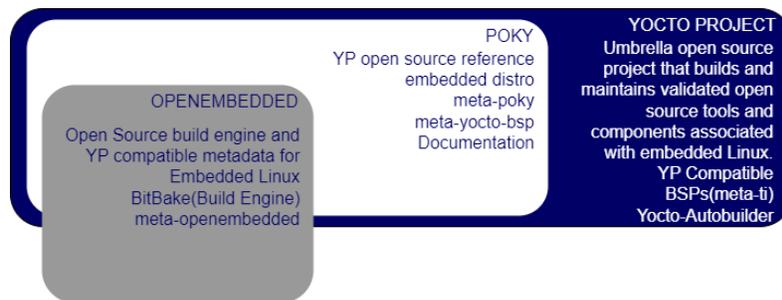


Figure 3:Yocto Project Cores

In our project, Embedded Linux distribution was prepared with the Yocto Project. In our project, Yocto Project was used to prepare Embedded Linux distribution, as it has a layer structure unlike other distribution preparation projects, keeps it up-to-date, and allows collaboration and reuse between commercial and open-source communities.

2.3.Python

Python is an interpreted, object-oriented, flexible, modular, high-level programming language with dynamic semantics.(Python Software Foundation, 2019). It was developed by Guido van Rossum and released in 1991.(w3Schools, 2019). English uses keywords regularly wherein other languages use punctuation marks such as semicolons, and is a high-level programming language that is simpler to analyze and understand than other programming languages with less syntactic shape than different languages (Python software program basis, 2019).

In our Platform Independent Embedded Linux OTA (PIELO) project, the OTA application was developed with the Python software language due to its ability to work in all Embedded Linux distributions regardless of the platform. Also, Python runs on an interpreter system, meaning the code can be run as soon as it's written. For this reason, the OTA application was developed with Python and the prototype was created quickly. In addition to being an interpreter, Python has been the preferred programming language in this project, thanks to its simple and high-level language structure, as one of the project's starting goals is easy development and easy use.

2.4.Cloud

Cloud is one of the most basic components in the OTA methods used. This cloud is a remote server that embedded devices can access. The software developed for certain reasons is first uploaded from the host computer to this cloud and becomes accessible to all devices in use. Also, since most remote servers can be used with a specific password, it ensures that new software is only available to eligible devices and provides security on the flow from the host to the embedded device.

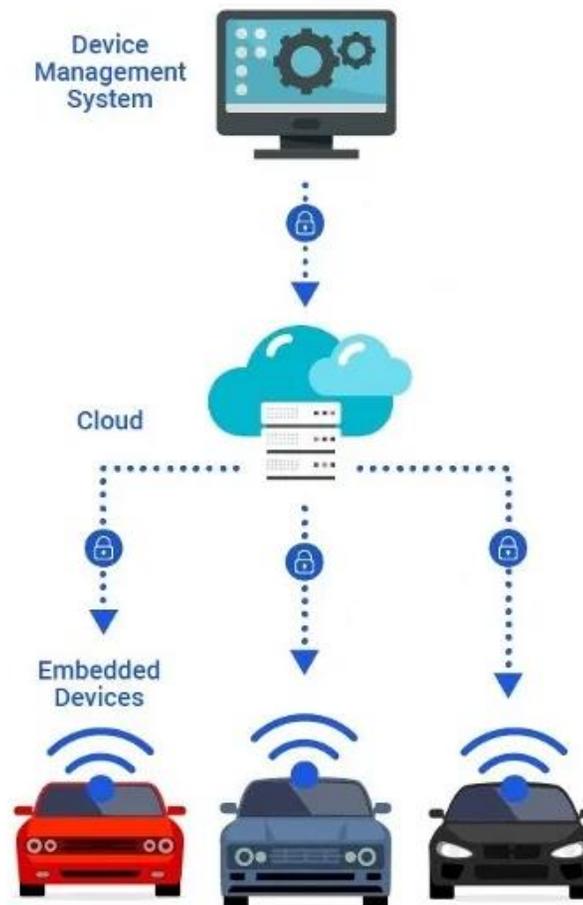


Figure 4: OTA process over the cloud (Koon, 2022)

Yocto images compiled during the development process of the platform-independent embedded Linux OTA method were versioned and Google Drive was used, considering that it is easy to use and access.

3. Method

There are open-source OTA mechanisms for updating embedded Linux devices. These mechanisms are platform dependent and difficult to use. Platform Independent Embedded Linux OTA was developed because of these drawbacks which are against our needs.

Platform Independent Embedded Linux OTA(PIELO) consists of Embedded Linux device firmware development, OTA application development, and Cloud Drive Configuration. At these stages, software development and testing activities were carried out.

3.1. Embedded Linux Device Firmware Development

An Embedded Linux distribution was created according to the requirements using the Zeus version of the Yocto Project.

NXP iMX8QM system-on-chip was used as the test platform. For this reason, an operating system compilation was made with Yocto for the iMX8QM SoC. The built operating system image is compressed in tar.bz2 format.

3.1.1. Bootlader Update

The NXP iMX8QM used as a test platform has ARM architecture. For this reason, U-Boot was used as the bootloader. U-Boot is the industry-leading and most used bootloader for Embedded Linux devices. Developed by DENX Software, Das U-Boot gave its first release in 1999. Released as open source under the GNU GPLv2 license. It offers support for a wide variety of architectures such as MIPS, ARM, AVR32 and x86 (Viswanathan, 2008).

The partition to flash the image that will run on the device has been determined and specified in U-Boot(Universal Boot Loader). The partition to which the Yocto image released with a more recent version will be flashed has been updated to the U-Boot source code using the C programming language. In the last step of the bootloader update, when the OTA procedure was completed, the partition that the device would mount and boot from was determined by the update made in the U-Boot source code, and the U-Boot was recompiled with these patches.

3.2. Cloud Configuration

The Embedded Linux image created with Yocto, compressed in tar.bz2 format, was uploaded to Google Drive, which was determined as the test server. It has provided more efficient use of image cloud drive stored in compressed file format.

According to the requirement, each cloud drive can be easily integrated into the system.

3.3. OTA Application Development

The Platform Independent OTA application was developed with Python as it can run on all Embedded Linux distributions.

The OTA application checks the firmware version in the cloud drive. It reads the version information engraved on the image running on the embedded Linux device and compares it with the firmware version in the storage. The application decides which version is more current. If the version of the firmware running on the Embedded Linux device is more recent, the application does not perform any operation and continues its control. If the firmware version in the storage area is more up-to-date, the application is downloaded from the cloud to the memory area specified in the compressed firmware device. After the download is complete, the compressed file is extracted and flashed to the specified memory location. The platform-independent OTA application performs all these operations by working in the background without interfering with the operating functions of the device in any way.

When the device reboots after the procedure is complete, the device will boot with the new version firmware, completing the device update.

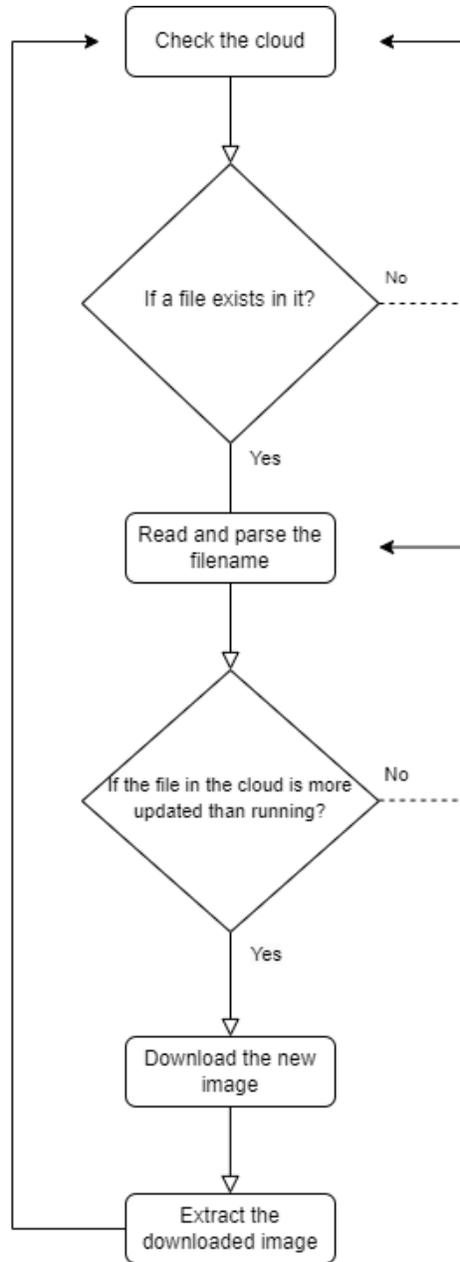


Figure 5:Flow chart of the OTA application

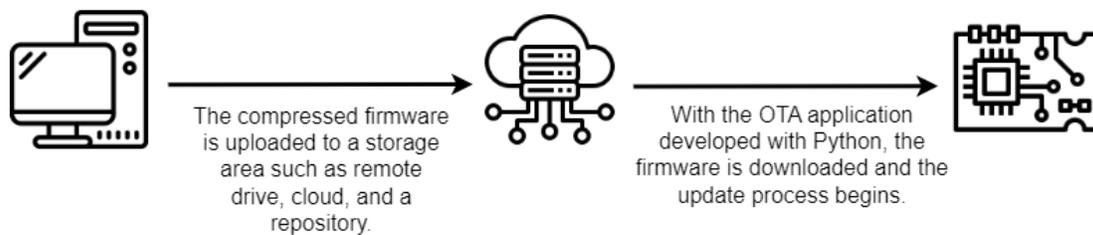


Figure 6:PIELO Basic Working Flow

4. Results

Platform-independent embedded Linux OTA method was performed with the mentioned platform by using the method given above. As a result of these studies and tests, our update method, which we aim to run independently of the platform, is much easier to use than other OTA methods on the market, and occupies a very small space in the memory of the embedded device, has become operational.

As an inference, if the materials used are desired to be changed, very small changes will be sufficient for this, and there will be no need for any procedure that will disrupt the flow. By way of illustration, the programming language that was used can be replaced with any language that can perform the same. It is enough to write a new code to replace them. Additionally, if the board (NXP i.MX8QM) is desired to be changed, it will be sufficient to specify the machine variable in the compilation of the Yocto project with the new platform name.

5. Discussion and Conclusion

To sum up, the PIELO OTA method was developed and can be used in projects with its key features such as ease to use, platform independency, and small footprint. The aimed features that were planned to get at the beginning of the project were provided. On the other hand, in addition to the desired basic features, there are still many aspects that are open to development such as security, power safe, and scheduling. Secure updating can be provided by encrypting the software with a crypto algorithm from the host machine to the embedded device. Moreover, a protection that will prevent the problem that may be experienced with the power cuts during the update and even continue the update from where it left off will be a feature that will carry this method further. Lastly, it can be made more useful by selecting the time period or condition to be updated by the end user.

6. Acknowledge

We are grateful to our mentor, Software Architect Hüseyn Karaçalı, for his encouragement and technical guidance.

Also, we thank TTTech Auto Turkey for their support to develop this project.

References

- [1] Brown, B.B. (2018). Over-the-Air (OTA) Updates in Embedded Microcontroller Applications : Design Trade-Offs and Lessons Learned.
- [2] Barbieri, E. (2021, November 24). What is embedded linux? part I. Ubuntu. Retrieved December 24, 2022, <https://ubuntu.com/blog/what-is-embedded-linux>
- [3] Wikipedia Contributors. (2019, June 29). Linux kernel. Wikipedia; Wikimedia Foundation. https://en.wikipedia.org/wiki/Linux_kernel
- [4] The Yocto Project. It's not an embedded linux distribution, it creates a custom one for you. Yocto project; Linux Foundation. Retrieved December 24, 2022, from <https://www.yoctoproject.org/>
- [5] What is the Linux kernel? (2019, February 27). [RedHat. https://www.redhat.com/en/topics/linux/what-is-the-linux-kernel](https://www.redhat.com/en/topics/linux/what-is-the-linux-kernel)
- [6] Python Software Foundation. (2019). What is Python? Executive Summary. Python.org; Retrieved December 24, 2022, from <https://www.python.org/doc/essays/blurb/>
- [7] w3Schools. (2019). Introduction to Python. W3schools.com. https://www.w3schools.com/python/python_intro.asp
- [8] Koon, J. (2022, December 1). Cybersecurity Risks Of Automotive OTA. Semiconductor Engineering. <https://semiengineering.com/cybersecurity-risks-of-automotive-ota/>
- [9] Viswanathan, K. (2008, February). How to setup a bootloader for an embedded linux machine. MSys Technologies. <https://www.msystechnologies.com/blog/how-to-setup-a-bootloader-for-an-embedded-linux-machine/>