Evaluation Analysis of Defect materials for IoT Embedded System

Kyu Tae Lee, Hyun Chang Lee, DoHyeun Kim

Abstract ***As the copyright disputes caused by illegal copy of IoT devices are increasing, it is required to judge the similarity to the embedded system such as Set-Top-Box and IPTV. However, due to the characteristics of the system, it is difficult to obtain the objective materials to evaluate the similarity. In this paper, we analyze evaluation items and the utility of results to determine the similarity in the case of difficulty in securing the materials in the dispute of the embedded system. Embedded systems figure out with hardware, software, and operating systems that work together. It is possible to set evaluation items such as circuit board configuration, connection pattern, signal input / output method, operation sequence, and GUI expression, even if there is no program source. Also, it is shown that the similarity between hardware and software can be verified as a result of comparison between both sides. The verification object is the essential data for judging the similarity of both sides, and it is an important data to perform the one-to-one comparison with the object. In this paper, we propose that evaluation items can be set by utilizing the characteristics of the system even when the object is insufficient, suggesting the possibility that it can be used as a useful proof for copyright dispute.***

Keywords: copyright, evaluation, comparison, ownership, embedded system, illegal copy

I. INTRODUCTION

Data mining helps to extract the original and the valuable data from the large amount of dataset. As the need for information devices connected to the Internet has expanded, the development of embedded systems in which hardware and software are interconnected is increasing. Systems based on OS(Operating system) are being used as service devices such as IPTV, STB(Set Top Box), and IoT(Internet Of Things), with the development of application programs. As a way to shorten system development time for various purposes, an illegal copy of a copyrighted product is reproduced and a dispute arises that infringes the original developer's rights. However, because alleged piracy claims to use an open-architecture technology, there is a need to judge the similarity between the original product and piracy.

For similarity evaluation, both sides should provide the data used in the development stage, but there is an unfair situation in the submission of data from the side with the doubt of illegal copy. Since embedded system products are manufactured by cross-development method, the products to be sold do not contain the program source code at the time of development, and only the executable files are stored in compressed form. [1] Therefore, if data of source code are not given, similarity should be judged only by products from both sides. In particular, this type of IT device may be developed by a new product as an outsourcer, and the client may not be the owner of the copyright and the client may become the original proprietor. Illegal copy of information devices is mainly caused by internal technicians participating in the development process moving to another company through turnover, producing the same products, producing and selling similar products, and causing losses. In this case, the developer who claims to be the original developer sues only those products that are suspected of being stolen without ownership of the development data of the device. In such a quarrel, when a similarity evaluation of the object is requested, the verification should be performed without providing the source code of the product or the drawing of the circuit board.

This study deals with the similarity analysis method and the analysis result when the development data corresponding to the object is not provided.

II. OBJECT MATERIALS BASIC

Most of the information devices used as portable terminals are made up of the processor-based IO(input and output) hardware and application software based on the OS(operating system). The operating system uses open source programs such as Linux Kernel, develops device drivers to enable hardware interfaces, and adds its own functions to the kernel[2,3]. The configuration of portable information devices is shown[Figure 1]. It contains a RAM used as a temporary storage for a program, a FLASH ROM for storing application programs, a keyboard used for input / output of information devices, an LCD and a touch pad, and a communication port for external communication (USB, COM).

Revised Manuscript Received on May 22, 2019.

Kyu Tae Lee, Div. ICE, Kongju National University,1223-24 Cheonan-Daero, Cheonan,31080,S.Korea

Hyun Chang Lee, Div. ICE, Kongju National University,1223-24 Cheonan-Daero, Cheonan, 31080,S.Korea

DoHyeun Kim, Dept. Of CE, Jeju National University,102, Jejudaehak-ro, Jeju-si, Jeju-do, 63243, S.Korea

E-mail:kimdh@jejunu.ac.kr.

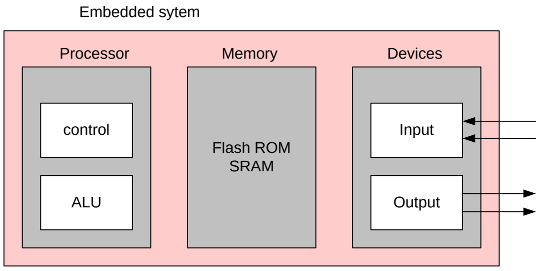


Figure 1. Embedded system architecture

Based on such a hardware configuration, the application program is created using a programming language familiar to the developer, and the completed program execution code is stored in the flash ROM. At this time, the application program utilizes various functions of the processor and peripherals (key pad, LCD monitor, LCD touch screen and communication port) to provide a terminal function useful to the user[4, 5]. As the function of the terminal is diversified and the function of the core processor is improved, the information device developed with the embedded structure is manufactured by installing a system management program known as an operating system (OS) as shown [Figure 2]. An operating system is software that has the ability to manage all functions of the system such as peripherals, memory, files, and interfaces with functions similar to those of a Personal computer.

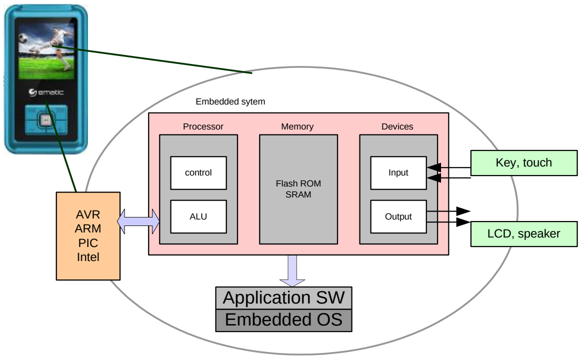


Figure 2. Device with Operating system

A device that operates the terminal in real time is called an embedded information device. Therefore, the software part developed is the kernel device driver, and the application program for the user interface and is added to the operating system.

III. EVALUATION ITEMS

The embedded information device is characterized in that an interface device, a device driver, and software including an application program are interlocked and operated based on a processor. Therefore, similarity comparison can be obtained by dividing the overall information of the system into items as follows.

1) Design of the whole system
2) PCB board
3) Schematic Circuit
4) Software source
5) Operating sequence

3.1. Design of the whole system

The system refers to the type of product to be handled by the user and it means the size, color, weight, position and shape of the interface button, size and resolution of the display screen. All of the elements known as 'product' specifications with the product case can be considered as evaluation items. This is because, from the point of view of developers who are developing new products, everything starts from the idea of the developer and is implemented. If a similarity is found in these elements, it can be judged that the pirate has unauthorized use of the developer's intellectual property rights[6, 7].

3.2. PCB Board

PCB (Printed Circuit Board) is a circuit board for connecting the design contents of a circuit diagram in practice. A copper board is made on a plate of a certain size, and a circuit component is connected. This includes the size of the PCB, the pattern of the connection pattern, the position of the component, and the spacing between components. Therefore, if similarity is shown in the above items, it can be regarded as a basis for the illegal copy.

3.3. Schematic Circuit

A circuit schematic is a circuit diagram showing the kinds of components constituting the hardware of the information device and the connection lines between the components. The circuit diagram includes the developer's proprietary technology such as the size position of the parts in the drawing, the lengths of the connection lines and connection lines between the component terminals, and the names given to the connection lines. And there are all developer's unique information. It is the basis for judging that the developer's copyright has been stolen.

3.4. Software Source

There is a difference in the language or operating system used depending on the processor applied to the system. When the operating system is applied, the structure of the kernel, the usage model, the device driver, and the application program are compared. Also, language and program flow chart used in development are used as an important software comparison elements[8, 9]. However, most of the defendant's data is poorly provided, and there are many cases where the defendant is not cooperating. Therefore, the similarity to the copyrighted subject matter claimed by the complainant is derived from the developer's point of view.

3.5. Operating sequence

The operating characteristic is referred to sequential steps such as a user's switch, a touch pad, and information input, which after power on, the information device is used and operated according to the display state. This process is designed by the developer considering the usability of the system. Based on this procedure, the program is created and the user function is implemented[10]. Therefore, even the case of the system or the appearance of the system is different, if the operation procedure is similar, the possibility of illegal copy may be suspected.

IV. TYPES OF EVALUATION

Disputes caused by illegal copy of information devices are caused by illegally utilizing the development technology of original copyright owners, producing similar systems, and releasing them to the market at low cost, as a result it makes economic losses of original developers. Evaluations are carried out to determine whether there has been a technical steal as a way to resolve a dispute, or whether the product is of a public range that anyone can develop using the open technology. As shown in [Figure 3], the cause of the dispute arises from the interested party, and the complainant makes a complaint to the investigating copyright authority. The requesting agency shall prove to both sides of the dispute the part corresponding to the cause of the economic damage or request the submission of the proof materials. When the evaluation materials are submitted, the compare items and the evaluation data are included in the professional copyright agency. At this time, if the core part of the interest can be identified, the copyright agency shall include a specific request for evaluation through agreement between the parties.

There are three types of copyright disputes consisting of systems. 1) the dispute that occurs when a development participant turns to another company and produces the same product. 2) the dispute that copyright owner commission other developers with technical skills to make products. However, this occurs when the developer produces and sells similar products. 3) The original proprietor commissioned a skilled developer to make the product. The dispute arises when the developer leaks technology to another company and produces a product from a third party.

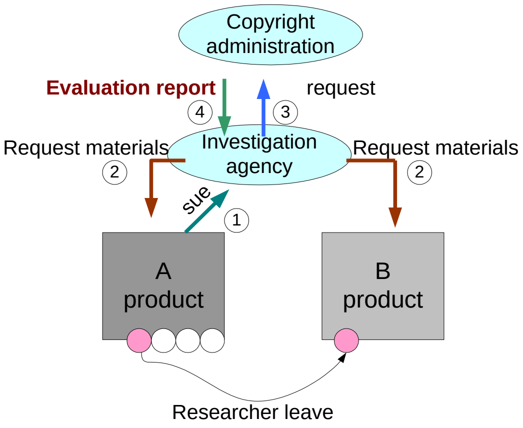


Figure 3. Flow of evaluation process to resolve dispute

In the first and second cases, the materials used for development and the source code of the program can be provided by both sides, and the object materials can be submitted. In the third case, the developer of the reproductions does not belong to the disputed company, and if the company is abroad, securing the object is limited to the product sold in the market. Therefore, in this case, it is difficult to perform the evaluation, because the program source code for the objectsis not provided.

Especially, since the information device including the operating system such as the embedded system is manufactured by the cross development method, the source code is embedded in the computer for development and the object system is implemented with the executable file such as STB, IPTV with IoT, etc. [Figure 4]. In this case, the source program in the development stage must be provided as a object materials, and similarity comparison is possible[12].

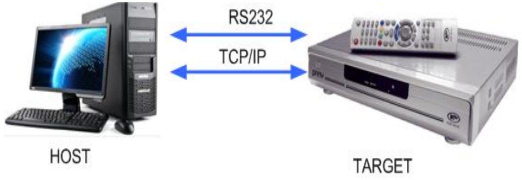


Figure 4. Environment of cross development for Embedded system

Embedded type evaluation object is a feature that is produced in the environment of cross development. Therefore, the source program data is not included in the user target system. Therefore, in order to judge similarity, the situation that both source codes are not submitted occurs. However, since the program source depends on the hardware environment and is designed to suit the component parts and functions used, it is possible to indirectly judge the operation sequence of the system such as the interface type, and the PCB(Printed Circuit Board) pattern of the circuit board. In addition, by comparing the GUI(Graphic User Interface) configuration and the operation pattern when using the remote controller or the touch screen, the similarity can be derived, which can help the overall meaningful comparison result of the system.

There is a remote control for the system to communicate with the outside. In general, the remote control uses the NEC Format method, and the signaling method is more useful than the line-coding method in which serial data is used as a method of minimizing errors[11]. As shown in the [Figure 5], 16-bit custom code (vendor identification code) and 16-bit control code are output continuously. Each 16-bit code is composed of 8 bits of information and 8 bits of inverted information, the control code is allocated to 8 bits, and the remote control type is configured to have 256 different functions.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

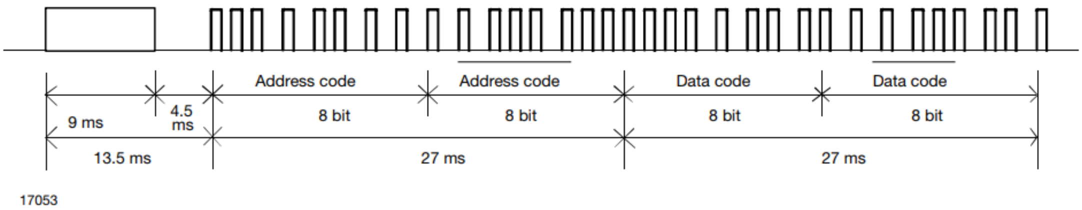


Figure5. NEC remote control format

At this time, in order to distinguish the logical value of the data to be continuously transmitted, there is a signal format for applying different pulse method to logic signals '1' and '0' as shown in the [Figure 6].

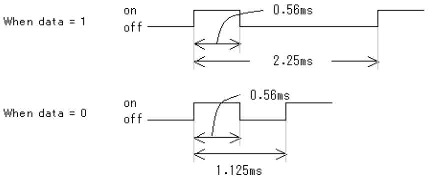


Figure 6. logic level ‘0’ and ‘1’ pulse define

Here, the custom code and the instruction code are arbitrarily set by the product designer and are implemented so that the promised function is operated in the receiver program. Therefore, since the functions defined in the code can not be the same for each product, even if there is no program source code of the corresponding device driver, it can be used as a data for judging the similarity of the two systems.The signals of the remote control of the IPTV used in general households are shown in the [Figure 7]. and there are the corresponding custom codes and instruction codes.

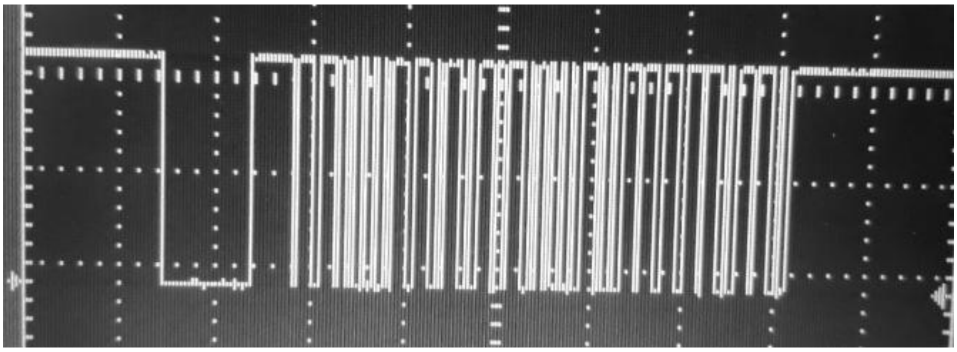
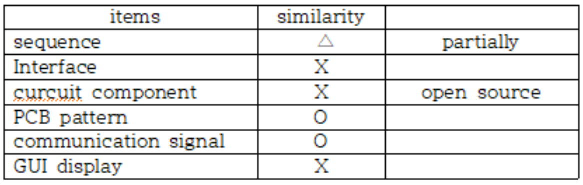


Figure 7. Remocon signal on receiver side

As the analysis shows, the custom code and control code of the remote control signal show [0x83] and [0x90] for 'power button'. If these remote control codes operate identically on other systems, it is possible to suspect that the remote control source code is the same. Generally, in the hardware and software design process of the remote controller, there is a feature that the developer arbitrarily assigns 256 functions of the control code and uses them. For this reason, most home appliances are manufactured with different remote control code values and are incompatible with each other. In other words, in the similarity evaluation when the program source code is not provided, the system interlocked with the hardware can be utilized for judging the similarity of the program by analyzing the proposed peripheral operation characteristic.

The following [Table 1] shows the similarity analysis results for the proposed evaluation items. Some similarities were observed between the operation sequence and the PCB pattern, and similar results were obtained with the remote control signal, and the circuit components showed no similarity by using the open technique.

Table 1: Evaluation result



items	similarity	
sequence	△	partially
Interface	X	
curcuit component	X	open source
PCB pattern	O	
communication signal	O	
GUI display	X	

I. CONCLUSION

The evaluation of the embedded system including the operating system may not be able to be compared with the source code only because the submission of the program source code that should be included in the object materials of the parties to the dispute is insufficient. However, in the characteristics of information devices, if the software are created based on hardware, the circuit diagram, operation sequence, and interface method constituting the system can be used as evaluation items. Therefore, by analyzing the items on both sides of the object materials, it is suggested that it can be used as the basis of similarity judgment. In addition, the source code of the program, including the operating characteristics of the system, should be included in the object for accurate similarity determination. In the future, in case of copyright disputes of the evaluation system type, we hope to complement the institutional policy that obliges the parties to submit the source program.

ACKNOWLEDGMENT

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT)(No.2018-0-01456, AutoMaTa: Autonomous Management framework based on artificial intelligent Technology for adaptive and disposable IoT), and this research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2019-2016-0-00313) supervised by the IITP(Institute for Information & communications Technology Planning & Evaluation). Any correspondence related to this paper should be addressed to DoHyeun Kim; kimdh@jejunu.ac.kr.

REFERENCES

1. Raj Kamal. Embedded systems Architecture Programming and Design. 2nd ed. MacGraw Hill Companies; 2015. p.5.(Book)
2. V.J. Mooney, D.M. Blough. A hardware-software real-time operating system framework for SoCs: IEEE Design & Test of Computers. 2002 Nov; 19(6): 44-51. (https://xploresit.ieee.org/document/1047743)
3. Kyu-Tae Lee, Hyun-Chang Lee, Jang-Geun Ki. Establishment of the Subtitle on Materials for Evaluating Intellectual Ownership: International Journal of Signal Processing, Image Processing and Pattern Recognition. 2017 Sep; 10(9): 79-88. (http://www.sersc.org/journals/IJSIP/vol10_no9/IJSIP%20ToC%20September%202017.pdf)
4. M. M. Swift, B. N. Bershad, and H. M. Levy. Improving the Reliability of Commodity Operating Systems: ACM Trans. on Computer Systems. 2003 Sep; 23(1): 77-110. (https://dl.acm.org/purchase.cfm?id=945466)
5. M. M. Swift, M. Annamalai, B. N. Bershad, and H. M. Levy. Recovering Device Drivers: ACM Trans. on Computer Systems. 2006 Nov; 24(4): 333-360. (https://dl.acm.org/citation.cfm?id=1189257)
6. G. Heiser. The Role of Virtualization in Embedded Systems: Proc. of First Workshop on Isolation and Integration in Embedded Systems. 2008 April; 11-16. (https://dl.acm.org/citation.cfm?id=1435461)
7. M. Rajagopalan, M. A. Hiltunen, T. Jim, and R. D. Schlichting. System Call Monitoring Using Authenticated System Calls: IEEE Trans. on Dependable and Secure Computing. 2006 July; 216-229. (https://www.computer.org/csdl/trans/tq/2006/03/q0216-abs.html)
8. W. Gu, Z. Kalbarczyk, R, K. Iyer, and Z. Yang. Characterization of Linux Kernel Behavior under Errors: Proc. of the 2003 Int. Conf. on Dependable Systems and Networks. 2003 Dec; 459-468. (https://pdfs.semanticscholar.org/c1a2/e9ae8f1de1a2d7057fb7bf26a5b0567c67de.pdf)
9. Compare files and folders [Internet], 2018 [updated 2018 Oct 10; cited 2018 Oct 10]. Available from: http://www.scootersoftware.com/features.php?zz=features_focused (website)
10. T. Naughton, W. Bland, G. Vallee, C. Engelmann, and S. L. Scott. Fault Injection Framework for system Resilience Evaluation: Proc. of the Resilience‟09. 2009 June; 23-28. (https://dl.acm.org/citation.cfm?id=1552530)
11. NEC code format[internet], 2018 [cited 2018 Oct 10]. Available from: https://www.vishay.com/docs/80071/dataform.pdf(website)
12. David E. Simon. An Embedded Software Primer . 12th ed. Pearson; 2005. p.191.(Book)