

논문 2021-1-4 <http://dx.doi.org/10.29056/jsav.2021.06.04>

임베디드 시스템의 동일기능 소스코드 유사도 분석 요구사항

김도현*, 이규대**†

Object Material Confirmation for Source Code Comparison on Embedded System

Do-Hyeun Kim*, Kyu-Tae Lee**†

요 약

임베디드 시스템 소스코드 감정목적물의 유사성을 판단하는 경우, 제공된 소스코드가 컴파일이 가능한 상태인지, 실행이 정상적으로 되는 것인지, 하드웨어와의 인터페이스가 일치하는지 등의 확인이 되지 않은 상태로 제공되는 경우가 지속적으로 발생하고 있다. 분쟁을 제기하는 측에서는 동작의 특성이 유사하고, 나타나는 기능의 효과가 유사한 이유로 소스코드의 많은 부분이 유사할 것으로 판단하고, 감정을 요청하게 되지만, 위의 여러 가지 상황으로 유사성 분석 결과가 기대와 다르게 나타나는 가능성이 우려된다. 본 연구에서는 감정사례를 통해 소스코드의 분석 과정과 검증되지 않은 소스코드의 유사성 도출의 개선방향을 제시한다.

Abstract

In case of evaluating the similarity of the source code analysis material in the embedded system, the provided source code must be confirmed to be executable. However, it is currently being in which compilation and interface matching with hardware are provided in an unconfirmed materials. The complainant assumes that many parts of the source code are similar because the characteristics of the operation are similar and the expression of the function is similar. As for the analysis result, the analysis result may appear different than expected due to these unidentified objects. In this study, the improvement direction is suggested through the case study by the analysis process of the source code and the similarity of the unverified source code.

한글키워드 : 소스코드, 목적물, 감정사례, 유사도, 라인비교

keywords : source code, object material, verification cases, similarity, line comparison

1. 서 론

* 국립 제주대학교 컴퓨터공학과

** 국립 공주대학교 정보통신공학부

† 교신저자: 이규대(email: ktleee@kongju.ac.kr)

접수일자: 2021.06.05. 심사완료: 2021.06.19.

게재확정: 2021.06.20.

임베디드 시스템으로 구성되는 제어에 사용되는 프로그램의 소스코드가 유사한 것으로 의심되어 분쟁이 발생한 경우, 감정 목적물로 양측의 소스코드가 제공되면, 코드의 라인단위 비교분석을 통해 유사성 검증이 가능한 특징이 있다[1].

이 경우 제공된 양측의 소스코드가 정상적으로 시스템에서 실행되는 것으로 검증된 경우에 해당된다. 그러나 현실적으로 소스코드 감정목적물의 비교 시, 제공된 소스코드가 컴파일이 가능한 상태인지, 실행이 정상적으로 되는 것인지, 하드웨어와의 인터페이스가 일치하는지 등의 확인이 되지 않은 상태로 제공되는 경우가 대부분이다. 이것은 도용의 의심을 갖는 피의자 측에서 의도적으로 정상적인 소스코드 제공을 거부하는 경우, 고소인 측에서 개발당시 자체개발을 못하고, 제3자에게 의뢰하여 개발함으로써 소스코드에 대한 정확한 자료가 없는 경우, 개발자가 퇴사나 이직으로 업체에 개발소스코드 자료가 부실하게 보관되어 있는 경우 등에 해당된다[2].

소스코드는 사용하는 언어의 종류, 사용하는 컴파일러의 종류, 컴파일러의 옵션 등에 따라 동일한 기능을 하는 소스코드라도 표현의 차이가 나타날 수 있으며, 이때 소스코드만의 유사성 분석결과는 만족한 결과를 얻지 못하는 경우가 나타날 수 있다.

피해를 주장하는 측에서는 동작의 특성이 유사하고, 나타나는 기능의 효과가 유사한 이유로 소스코드의 많은 부분이 유사할 것으로 판단하고, 감정을 요청하게 되지만, 위의 여러 가지 상황으로 유사성 분석 결과가 다르게 나타나는 결과를 얻을 가능성이 많아진다.

감정인의 입장에서 정확한 분석이 가능하기 위해서는 무엇보다 개발과정의 소스코드와 컴파일러 검증, 하드웨어 시스템과의 연동 검증 등이 확인된 목적물이 제공되었을 때, 소스코드의 비교와 컴파일러의 디버깅 기능을 활용한 검증을 통해 소스코드가 상이한 경우 기능분석을 통한 유사성 검증에 정확성을 도출할 수 있다.

본 연구에서는 감정사례를 통해 소스코드의 분석 과정과 소스코드의 유사성 도출의 결과를 분석하고, 개선점을 제시하였다.

2. 임베디드 소스코드 목적물

임베디드시스템과 연동되는 프로그램 소스코드는 개발단계에서 회로도를 작성하는 것으로 시작된다. 하드웨어를 구성하는 회로도가 작성되고, PCB작업으로 회로기판과 케이스 및 입출력 인터페이스 장치가 완성되면, 시스템의 기능에 해당하는 동작 흐름도를 작성하고, 이 흐름도에 따라 소스프로그램 코딩이 진행된다. 그림 1과 같이 기능흐름도가 작성되고, 코딩언어가 지정되면, 적합한 컴파일러를 사용하여, 프로그램 생성과 디버깅을 반복하면서 실행코드로 완성된다. 실행코드는 다운로드 과정으로 시스템의 펌웨어와 응용프로그램을 MCU에 옮겨지고, 이후부터 시스템은 독자적으로 동작하는 제품으로 완성되어 사용자에게 제품으로 출시된다[3][4].

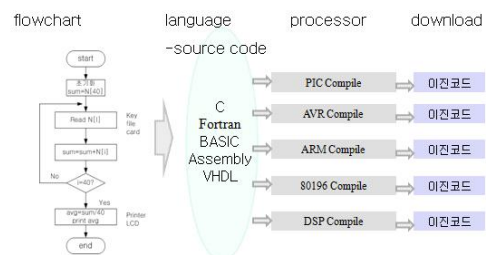


그림 1. 시스템 프로그램 작업 흐름도
Fig.1. Flowchart of system program

소스코드 목적물은 프로그램언어(C, Fortran, python 등)로 작성된 텍스트 형태의 파일을 지칭하는 것으로, 정상적인 목적물이 되기 위해서는 개발단계에서 컴파일 오류가 없이 진행되는 파일이어야 한다.

본 연구에서의 감정사례는 그림 2와 같이 B사에서 제품을 기획하고, 프로그램 개발 작업을 A사에 의뢰하였으나, A사의 사정으로 개발기간이 지연되었고, B사는 A사의 개발결과를 신뢰하지

않게 되었다. B사는 다른 C개발사에 프로그램개발을 의뢰하고 협력하여, 완성된 소스코드를 제품으로 판매하고 있는 있다. 결국 A사는 자사가 제공한 반제품의 소스코드 일부를 B사가 무단으로 도용하였다고 의심하여 A사의 소스코드와 B사의 소스코드의 유사성을 분석하는 감정을 요청한 목적물이다. 또한 시스템적으로 동일한 하드웨어환경에서 동작되도록 개발된 소프트웨어의 특성상 하드웨어부분의 센서 인터페이스관련 부분에서 유사한 소스 라인 부분이 적용될 수 있는 가능성이 있다. 하드웨어적으로 적용한 CPU가 특정되면, 해당 프로세서의 제작사에서 개발자의 지원을 위해 공개적인 인터페이스 프로그램이나, 특정 센서의 구동이 가능한 프로그램 소스를 제공하는 경우도 있다. 감정사례의 시스템이 동일한 하드웨어 환경에서 운영되는 특성으로 유사한 소스코드 부분이 확인될 가능성이 있다[5].

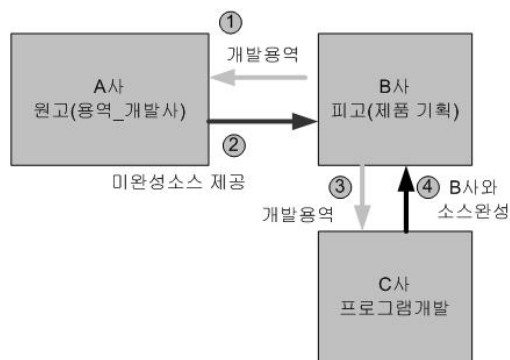


그림 2. 순서별 분쟁진행
Fig. 2. Sequence of dispute program

목적물의 확보 상황에서 발생하는 현실적인 문제점으로는 감정목적물 특정 시점에서 분쟁 대상 업체의 개발자가 퇴사나 이직 등으로 부재하거나, 개발과정에서의 문서가 작성되지 않아 소스프로그램 모듈에 대한 설명서나 기능에 대한 명세서가 보관되어 있지 않는 경우가 많고, 이때

소스코드나 실행코드만 제공되는 경우도 발생하고 있다[6].

본 사례에서는 소스코드가 제공되었지만, 프로그램의 사양서나 명세서가 보관되지 않은 상태로 전달되었다. 감정목적물은 A사가 보관중인 소스코드(A), A사로부터 B사가 개발 중에 제공받은 소스코드(B), B사가 C사와 협력하여 완성한 소스코드(C)의 3가지 소스코드가 감정대상으로 특정되었다. 그러나 C 코드는 B사와 C사가 협력 개발한 소스코드로 C사가 개발한 소스코드부분은 분쟁에 포함되지 않은 것으로 결정되어, C코드는 B사가 개발한 것으로 주장하는 소스코드가 제공되었다.

3. 소스코드 유사성

소스코드의 유사성 분석은 텍스트 문서의 유사성 분석과 유사한 방법으로 수행되며, 프로그램의 특성상 소스코드 문서의 첫 라인에서 시작하여 아래로 순서적으로 수행되기 때문에 라인단위로 분석된다. 분석 대상으로 하는 파일의 이름이 동일 유사하거나, 동일한 파일수로 구성되어 있는 경우에는 일대일 비교방법을 적용하며, 이때 beyondcompare 와 같은 상용분석 도구를 사용한다[7].

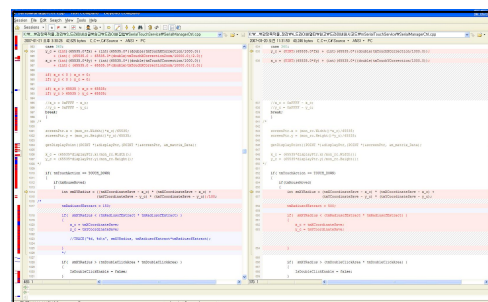


그림 3. Beyond Compare 실행 화면
Fig. 3. Beyond Compare window

그러나 운영체제의 커널과 같이 파일의 이름과 개수가 다른 경우에는 일대일 비교 대상을 특정할 수 없게 되어, 다대다 비교방법을 적용하게 된다. 다대다 비교방법은 모든 파일에 대해 비교 결과를 얻어 가장 유사한 파일들을 선정하고, 세부 비교방법으로 유사성을 도출한다[8][9].

목적물은 원본A와 비교본C에 대해 수행되었으며, 각 폴더에 포함된 파일마다 구성된 라인수를 확인하고, 비어있는 라인은 제거하여, ‘공백제거줄수’로 정제한 결과로 비교분석을 진행하였다.

비교도구는 양측의 파일의 수가 다르고, 파일 이름이 매칭되는 개수가 적어서, 그림 4와 같이 ‘다대다’ 비교가 가능한 exEyes를 사용하였다.

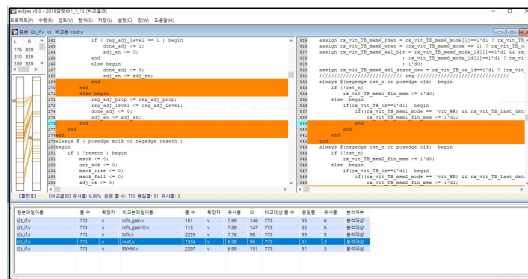


그림 4. exEyes v5.0 실행화면
Fig. 4. exEyes v5.0 window

4. 유사성 분석

목적물은 A소스코드의 경우 5개의 폴더에 파일이 분산되어 있으며 총 72개 파일로 구성되어 있고, 공백을 제거한 상태의 총 라인수는 30,455개로 확인되었다. C소스코드의 경우 단일 폴더에 36개의 파일이 포함되어 있으며, 공백을 제거한 상태의 총 라인수는 103,223개로 확인되었다. 특이 사항은 C 소스코드의 경우, 시스템을 구성하는 전체 파일일 중에서 B사의 개발부분으로 주장하는 부분만 감정목적물로 제출된 것이다.

유사성 검출 도구(exEyes)를 사용하여 비교분석 결과는 표1과 같이 동일 또는 유사한 것으로 추출된 3,615 라인이 확인되어, 11.8%의 유사성을 갖는 것으로 분석되었다.

표 1. 라인단위 원본-비교본 비교결과
Table 1. Comparison result with line to line

폴더명	A 소스코드		C 소스코드		유사줄수	분석결과 (%)
	파일 수	공백 제거	파일 수 (36개)	공백 제거		
folder1	27	9,125	-	38,721	980	10.7
folder2	8	5,480	-	13,424	702	12.8
folder3	7	4,956	-	10,907	676	13.6
folder4	8	4,439	-	13,101	746	16.8
folder5	22	6,455	-	27,070	511	7.9
합계	72	30,455	-	103,223	3,615	11.8

추가적으로, 유사한 것으로 분석된 부분의 소스코드의 라인 상황을 세부적으로 확인하기 위해 각 파일에 대한 유사추출 부분을 직접 확인하는 작업을 수행하였다. 육안으로 분석한 결과는 그림 5와 같이 대부분의 유사라인이, 프로그램의 언어 특성으로 나타나는 중괄호({}), 명령문(if, else)등이 하나의 라인을 점유함으로써 나타나는 중복 현상으로, 유의미한 유사성으로 판단하기 어려운 것으로 확인되었다. 또한 분석 대상 라인이 10만 라인 분량으로 이러한 결과가 누적되어 11%의 수치로 나타난 것으로 판단되었다.

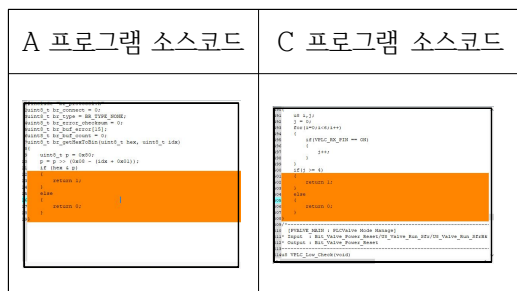


그림 5. 유사라인으로 추출된 부분의 육안비교
Fig. 5. Direct comparison to similar lines

결과적으로 11.3%의 유사성 분석 결과도 유의미한 내용이 포함되지 않는 것으로 판단되어, 전체적으로 분석결과는 유사성이 없는 것으로 판단된 결과를 보였다.

본 연구사례는 제공된 목적물이 전체 소스코드로 볼 수 없는 부분과, 제공된 양측의 소스코드가 컴파일이 성공적으로 수행되고, 하드웨어 제어보드로 다운로드 되어, 전원 연결 시 정상적으로 동작하는 과정이 확인되지 않은 상태로 분석된 결과이다. 감정목적물의 전체적인 확보 및 부분적인 목적물의 경우 양측의 검증과 동의가 확인되는 절차가 포함되어야 객관적이고, 정확한 분석결과를 도출할 수 있을 것이다.

5. 결 론

소스코드를 목적물로 수행하는 유사성 분석 작업에서 감정인은 양측의 소스코드가 실행가능하고, 분쟁 시스템에 적용되는 소스코드라는 가정 하에 분석을 시작한다. 본 연구의 감정 사례에서 분석한 결과와 같이 동일한 언어와 유사한 기능의 제어를 하는 소스코드에 대한 유사성 결과가 낮게 도출되는 경우에 목적물에 대한 검증이 미완으로 남는다. 실제로 해당시스템에서 실행되는 소스코드와 같은 것인지에 대한 검증이 없는 상태로 분석결과를 완료하게 된다. 결과적으로 감정인의 입장에서 정확한 분석이 가능하기 위해서는 무엇보다 개발과정의 소스코드 확보와 컴파일러실행에 의한 검증 및 하드웨어 시스템과의 연동 검증 등이 확인된 목적물이 제공되도록 하는 제도상의 보완이 요구되며, 이러한 과정을 통해 양측의 공정한 분석 결과 제공 및 분쟁 예방 효과가 있을 것으로 기대된다.

감사의 글

이 논문은 2020년도 정부(교육부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 연구임.(2020R1I1A3070744)

참 고 문 헌

- [1] Rajesh K. Gupta, "Introduction to Embedded system", ICS 212, 2002 winter workshop
- [2] 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, 10(9), pp.79-88, Sep. 2017. http://www.sersc.org/journals/IJSIP/vol10_no9/IJSIP%20ToC%20September%202017.pdf
- [3] Do-Hyeun Kim, Kyu-Tae Lee, "Management of Reliability and Delivery for Software Object Material", Journal of Software Assessment and Valuation (JSAPV), vol.15, No.2, pp.51-57. Dec. 2019. <http://dx.doi.org/10.29056/jsav.2019.12.07>.
- [4] 이규대, "유사성 비교에서 세부항목 설정 기준", 한국소프트웨어감정평가학회 논문지, 12권 1호, pp.21-26, June, 2016.
- [5] 이규대, "임베디드시스템의 이진코드 추출 및 분석", 한국소프트웨어감정평가학회 논문지, 5권 1호, pp.27-38, May, 2009.
- [6] 이규대, 권기영, "정보기기 감정에서 세부항목 설정 사례", 한국소프트웨어감정평가학회 논문지, 12권 2호, pp.9-14, Dec. 2016.
- [7] Compare files and folders [Internet], 2018 [updated 2018 Oct 10; cited 2018 Oct 10]. http://www.scootersoftware.com/features.php?zz=features_focused (website)

- [8] Raj Kamal, Embedded systems Architecture Programming and Design, 2nd ed., MacGraw Hill Companies, p.5, 2015. ISBN: 0-07-049470-3
- [9] Ali, Safdar, and DoHyeun Kim, "Building power control and comfort management using genetic programming and fuzzy logic", Journal of Energy in Southern Africa, 26.2, pp.94-102, 2015. http://www.scielo.org.za/scielo.php?script=sci_arttext&pid=S1021-447X2015000200010&lng=en&nrm=iso
- [10] 김도현, 이규대, "실행코드 비교 감정에서 주변장치 분석의 유효성", 한국소프트웨어감정평가학회 논문지, 16권1호, pp.37-44, June, 2020. <http://dx.doi.org/10.29056/jsav.2020.06.05>.

저 자 소 개



김도현(Do-Hyeun Kim)

2000.8 경북대 전자공학과(정보통신전공) 박사
2004.9~현재 국립 제주대학교 공과대학 컴퓨터공학전공 교수.

<주관심분야> 사물인터넷, 예측 및 최적 제어, 모바일 컴퓨팅, 임베디드 소프트웨어



이규대(Kyu-Tae Lee)

1991 고려대 전자공학과 박사
1992~현재 국립공주대학교 정보통신공학부 교수

<주관심분야> 신호처리, VLC, 저작권보호, 임베디드 시스템, 상황인식 및 학습