

기반 소프트웨어 프레임워크 CLI 서비스를 활용한 내장형 소프트웨어의 효율적인 개발 방안

Efficient Development Method for Embedded Software Using Next Software Framework CLI Service

이동언^{*.1)} . 김지원²⁾ . 진정훈³⁾ . 조규태⁴⁾

Dongeon Lee^{*.1)} . Jiwon Kim²⁾ . Junghun Jin³⁾ . Kyutae Cho⁴⁾

[초 록]

내장형 소프트웨어(embedded software)는 범용 소프트웨어가 동작하는 환경과 다르게 탑재 하드웨어가 제한되는 경우가 많다. 특히 내장형 소프트웨어가 동작하는 하드웨어 환경에서는 문자를 출력할 수 있는 화면이 없을 수도 있고, 키보드나 마우스와 같은 입력장치가 없거나 로그를 저장할 수 있는 저장장치도 제한적인 경우가 많다. 이러한 환경에서 내장형 소프트웨어를 개발하는 것은 일반 PC나 서버 환경에서 개발하는 소프트웨어에 비해 개발이나 테스트의 속도와 편의성이 차이가 많이 날 수 있다. 이와 같은 문제로 본 논문에서는 기반 소프트웨어 프레임워크(NSFW : Next Software Framework)의 Cross Platform CLI(Command Line Interface) 서비스를 활용하여 내장형 소프트웨어의 보다 효율적인 개발 방안을 제시한다.

[ABSTRACT]

Since the embedded software is designed to reduce the production cost by optimizing the hardware to focus on a specific purpose, there are many cases where the hardware to be mounted is limited differently from the environment in which general-purpose software operates. In particular, in a hardware environment in which embedded software is operated, there may be no screen to output characters, there are no input devices such as a keyboard or mouse, or storage devices capable of storing logs are often limited. Developing embedded software in such an environment may differ greatly in speed and convenience of development or testing compared to software developed in a general PC or server environment. Due to this problem, this paper proposes a more efficient development plan for embedded software by utilizing the Cross Platform CLI(Command Line Interface) service of the NSFW (Next Software Framework).

Key Words : Embedded software(내장형 소프트웨어), Cross Platform(크로스 플랫폼), CLI(명령 줄 인터페이스), NSFW(기반 소프트웨어 프레임워크)

1. 서 론

내장형 소프트웨어(Embedded Software)는 일반적으로 하드웨어 및 소프트웨어로 구성된 제한된 컴퓨팅 환경에서 특정 기능을 실행하도록 구성이 된다.

비디오 플레이어, 텔레비전 및 휴대폰과 같은 안전이 치명적이지 않은 시스템에서부터 항공기, 자동차 및 의료 장비와 같은 안전이 중요한 시스템에 이르기까지 다양하고 복잡한 응용 프로그램을 처리하는데 사용된다^{1,2)}. 특히 국방 무기체계 분야에서 전자전, 유도무기, 전투체계 등 내장형 소프트웨어를 적용하여 개발하는 시스템이 많다. 이러한 내장형 소프트웨어는 생산 비용을 줄이도록 특정 목적에 맞추어서 하드웨어를 최적화하여 설계하기 때문에 범용 소프트웨어가 동작하는 환경과 다르게 탑재 하드웨어가 제한되는 경우가 많다³⁾. 특히 내장형 소프트웨어가 동작하는 하드웨어 환경에서는 문자를 출력할 수 있는 화면이 없을 수도 있고, 키보드나 마우스와 같은

1) LIG넥스원 지능형SW연구소(Intelligent Software System, LIG Nex1, Korea)

* Corresponding author, E-mail: dongeon.lee@lignex1.com

Copyright © The Korean Institute of Defense Technology

Received : August 21, 2021

Revised :

Accepted : September 9, 2021

입력장치가 없거나 로그를 저장할 수 있는 저장장치도 제한적인 경우가 많다^{4,5)}. 이러한 환경에서 내장형 소프트웨어를 개발하는 것은 PC(Personal Computer)나 서버 환경에서 개발하는 소프트웨어에 비해 개발이나 시험의 속도와 편의성이 차이가 많이 날 수 있다. 특히 다품종 소량생산이 많은 방위산업의 특성상 다양한 동작환경이 있을 수 있는데 이러한 상황에서 크로스 플랫폼(Cross Platform)이 가능해 모든 환경에서 동일한 명령어와 동일한 결과를 얻을 수 있다면 보다 효율적인 개발 및 시험이 가능해진다. 이와 같은 문제로 본 논문에서는 기반 SW 프레임워크(NSFW: Next Software Framework, 이하 NSFW)의 Cross Platform CLI(Command Line Interface) 서비스를 활용하여 내장형 소프트웨어의 보다 효율적인 개발 방안을 제시한다.

2. 내장형 소프트웨어의 개발 연구

본 절 1에서 언급한 것과 같이 일반적으로 내장형 소프트웨어가 가지고 있는 특징들을 고려하여, 효율적인 내장형 소프트웨어의 개발을 위해 원격 접속을 통한 소프트웨어 모니터링, 소프트웨어 제어 기능 등을 개발한다. 본 기능은 내장형 소프트웨어에서 공통적으로 사용될 수 있는 기능이므로 다양한 활용을 위해 LIG넥스원에서 공통적으로 사용되는 기반 SW 프레임워크(NSFW)에 포함시켜 개발하도록 한다.

2.1 기반 SW 프레임워크(NSFW)

기반 SW 프레임워크(NSFW)는 LIG넥스원에서 개발하는 소프트웨어에 공통적으로 탑재되는 기반 프레임워크이다. Fig. 1은 기반 SW 프레임워크(NSFW)의 구성도를 간략하게 보여준다. 다양한 OS에서 동작이 가능한 프레임워크이며 10가지의 서비스(Framework, Task, Sync, IPC, Timer, Log, Dynamic Module, CLI, Message, Test)를 포함하고 있다. 본 논문에서는 NSFW가 제공하는 10가지의 서비스 중 내장형 소프트웨어 개발 및 시험에 도움을 줄 수 있는 Cross Platform CLI 서비스에 대해 설명한다. 그리고 Fig. 2는 기반 SW 프레임워크(NSFW)의 설계에 대한 보다 자세한 구조를 보여준다. 기반 SW 프레임워크(NSFW)는 크게 2개의 층(Layer)으로 구성되어 있다. 운영체제(Operating System)와 긴밀하게 연동하여 각

운영체제에 맞는 기능들을 수행하는 OSA(Operating System Adaptation) 층이 아래의 층에 위치하고, 운영체제와 독립적으로 사용자에게 각 서비스를 제공하는 서비스(service) 층이 윗 층에 위치한다. 이와 같은 구조를 통해서 차후 운영체제가 추가되거나 변경되더라도 서비스 층에 변화를 최소화할 수 있다. 그리고 사용자에게 공개하는 API(Application Programming Interface)는 변경하지 않을 수 있어 크로스 플랫폼(Cross Platform)이 가능해진다¹⁶⁾.

2.2 Cross Platform CLI에 필요한 기능

OS와 독립적으로 CLI 기능이 동작할 수 있도록 OS와 서비스층 사이에 위치하도록 설계된 부분인 OSA(Operating System Adaptation)는 각 OS에 맞게 기본 기능들이 쓰레드, 파일, 타이머, 타임, 통신 등을 각각 설계 및 구현이 되어야 한다. 그리고 운영체제와 독립적으로 동작하는 기능인 Cross Platform CLI에 필요한 기능을 크게 7가지로 구분하였다. 각 기능은 아래 표 1과 같다. CLI 명령을 처리하는 부분인 show와 change, etc의 경우 다양한 세부 기능들을 포함하고 있다. 각 데이터들은 예를 들어 NSFW가 제공하는 기능에서 사용하는 데이터를 활용하여 나타내었으며, 각 데이터는 노드, 태스크, 통신, 타이머, 로그 등을 포함하고 있으며 이러한 데이터를 전시하거나 변경 및 조작할 수 있는 기능, 기타 기능들을 포함한다.

표 1. Cross Platform CLI에 필요한 기능 목록

Table 1. List of Features Required for Cross Platform CLI

이름	설명
connect	원격으로 CLI 서비스에 연결하는 기능
show	show(전시) CLI 명령을 처리하는 기능
change	change(제어) CLI 명령을 처리하는 기능
restart	restart(재시작) CLI 명령을 처리하는 기능으로 수행시 응용 소프트웨어가 재시작 됨.
monitor	monitor(모니터링) CLI 명령을 처리하는 기능으로 데이터의 상태를 실시간으로 전시한다.
prt	prt(프린트) CLI 명령을 처리하는 기능으로 수행시 특정 로그의 출력을 CLI를 통해 전시한다.
etc	etc(기타) CLI 명령을 처리하는 기능.

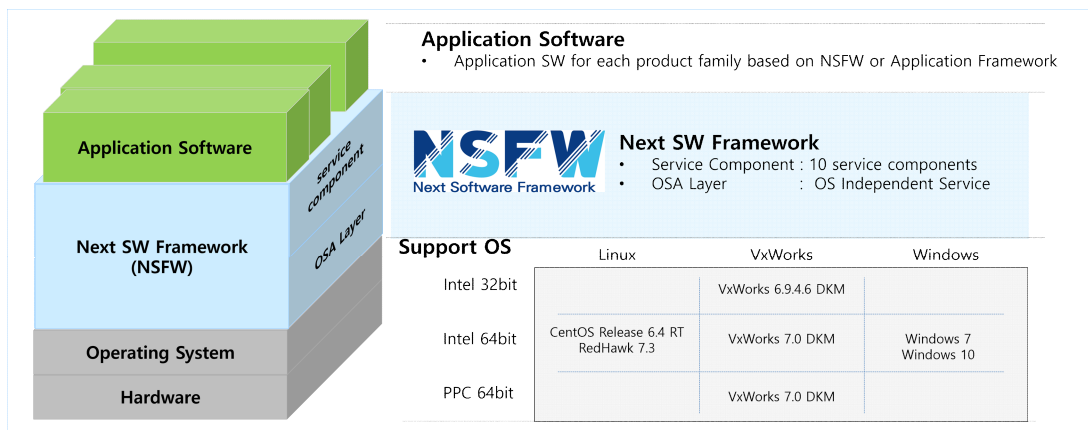


그림 1. NSFW 구성도

Fig. 1. Next SW Framework (NSFW) architecture

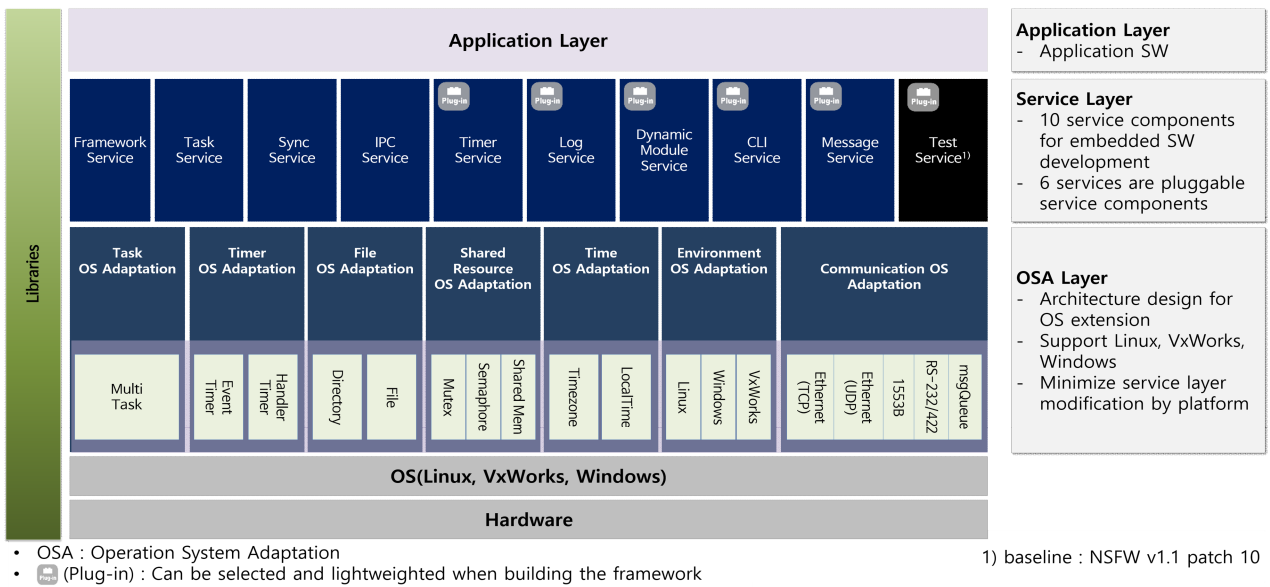


그림 2. NSFW 구조

Fig. 2. Next SW Framework (NSFW) architecture

Fig. 3은 화면이 없는 임베디드(Embedded) 환경에서 효율적으로 개발할 수 있도록 텔넷(Telnet)을 통해서 원격으로 명령을 수행할 수 있는 방안을 간략히 보여준다. 앞서 언급한 OSA를 통해서 사용자는 지원하는 다양한 운영체제에서 모두 동일한 경험을 할 수 있는 점이 특징이다. 이를 위해서는 각 OS에 특화되어있는 부분을 따로 구현해 주어야 하고 또한 표준화되어있는 부분은 ANSI(American National Standards Institute)와 같은 표준 라이브러리와 API를 사용하고 구현해야 한다. 그리고 Telnet에 대한 접근방법을 일반적으로 사용하는 이더넷(Ethernet)뿐만 아니라 임베디드 환경에서 하드웨어적으로 빈번하게 제공하는 RS-232/422과 같은 시리얼(Serial) 통신을 통해서도 동일한 기능을 손쉽게 사용할 수 있도록 개발해야 한다.

2.3 Cross Platform CLI에서 필요한 세부 기능
 본 절 2.2 Cross Platform CLI에 필요한 기능들에 대해 보다 자세하게 살펴본다.

2.3.1 connection telnet

먼저 원격으로 Cross Platform CLI 서비스에 텔넷(telnet)으로 연결을 처리하는 기능이 필요하다. 개발자가 디버그(debug) 모드로 컴파일(compile)한 응용 소프트웨어에 특정 포트(port)를 설정 파일에 지정한 다음 telnet으로 접속할 수 있다. 보안을 위해 지정된 포트에 접속시 비밀번호 입력을 통해서 접속 사용자를 제한한다. 특히 보안이 매우 중요한 국방 소프트웨어의 경우에는 이러한 접속을 원천적으로 차단하여 보안상 위험을 최소화하기 위해 릴리즈(release) 모드에서는 동작하지 않도록 한다. 그리고 접속에 성공하면 개발자는 프롬프트를 이용해서 명령을 Cross

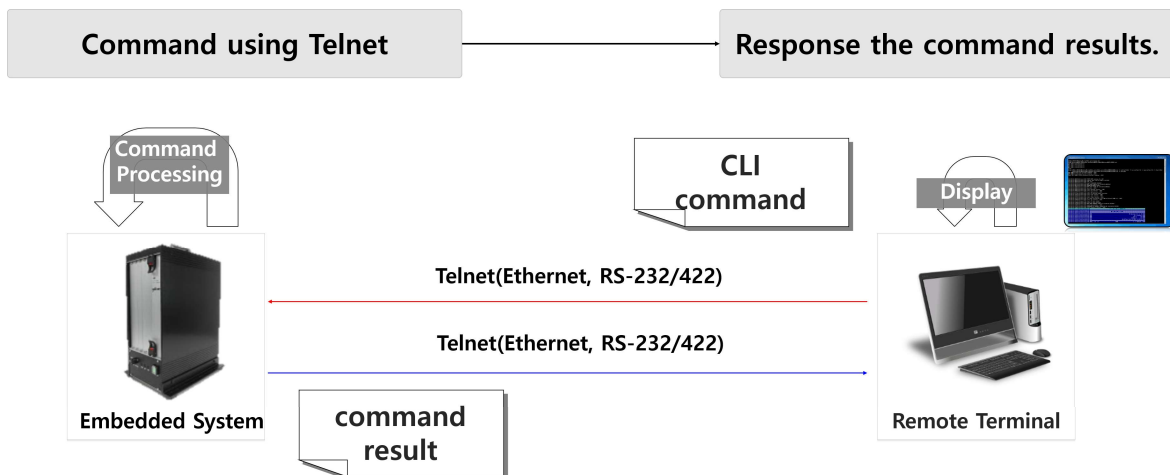


그림 3. 화면이 없는 임베디드 환경의 효율적인 개발 방안

Fig. 3. Efficient development plan for embedded environment without screen

Platform CLI에 보낼 수 있는 입력가능 상태가 된다. 개발자는 이 상태에서 다양한 명령어를 요청할 수 있으며, 모든 입력행위를 마치고 나서 개발자가 명령어(q)로 종료를 요청하면 CLI 서비스는 처리한 데이터를 모두 해제하고 텔넷 연결을 끊는다.

2.3.2 process CLI command (show)

Cross Platform CLI에 접속 및 인증이 성공한 상태인 입력가능 상태에서 show 명령어를 이용해 시스템이 제공하는 데이터를 전시할 수 있는 기능이 필요하다. 기본적으로 내장형 소프트웨어가 제공하는 서비스들의 현재 상태를 확인할 수 있으며, 본 논문에서는 NSFW의 데이터를 예로 들어 설명한다. CLI 서비스에서 제공하는 정보들은 응용 소프트웨어 이름, 버전, 컴파일 날짜, 설정파일 정보, CLI 포트정보, 엔디안(endian), 운영체제 종류, 운영체제 비트(bit) 등을 포함하는 프레임워크 정보부터 노드(node) 관리를 위한 마스터(master), 슬레이브(slave) 등을 포함하는 노드정보들, 각종 통신(TCP, UDP, Serial, MIL-STD-1553B 등)들의 정보, 태스크(task) 정보, 타이머(timer) 정보, 메시지 기반 통신을 제공하는 메시지(message) 정보, 태스크의 역할들을 나눠서 분할 및 구조화한 루틴(routine) 정보, 태스크간 동기화를 위한 리소스(resource) 정보, 로그(log) 정보까지 다양한 현재 상태를 제공한다. 내장형 소프트웨어 개발 및 시험을 할 때 구동 하드웨어 환경에서 화면이 없거나 저장매체가 없어 실제 동작 중인 응용 소프트웨어와 프레임워크의 정보들을 확인하기 어려울 때 이 기능을 사용하면 편리하게 해당 정보를 얻을 수 있다.

그리고 조금 더 자세한 정보를 확인할 수 있는 기능도 필요하다. 예를 들어 태스크(task)의 세부 정보를 볼 수 있는 기능은 특정 태스크의 정보를 전시하는 화면을 보여준다. 일반적으로 실제 동작중인 태스크를 확인하기 위해서는 추가적으로 제 3의 소프트웨어를 사용하거나 특정 운영체제에 동작 가능한 디버깅 툴을 이용해야하는데 내장형 소프트웨어에서는 해당 방법을 택하기 쉽지 않은 경우가 많다. 이는 내장형 소프트웨어는 저장매체나 화면과 같은 외부 디바이스의 연결이 제약된 경우가 많기 때문이다. 이러한 상황일 때 Cross Platform CLI를 사용하여 특별한 추가 장비나 소프트웨어 없이 현재 동작중인 태스크의 자세한 정보(할당된 CPU 코어정보, 코어마스크, 스택사이즈, 스케줄링 정보, 우선순위, 동작시간, CPU 사용량 등)를 확인할 수 있다면, 개발이나 시험에서 도움을 줄 수 있다.

이러한 정보들은 각 시스템에 맞게 손쉽게 추가 및 변경이 가능하도록 설계되어야 하며, 각 정보들은 정형화된 구조에 맞게 저장되어야 관리가 용이하다.

표 2는 Cross Platform CLI에서 필요한 show 기능에 포함되는 명령어들을 NSFW의 데이터를 기준으로 전시(show)하는 기능으로 나열하였다. 크게 NSFW의 프레임워크 정보와 응용 소프트웨어 정보로 나눌 수 있으며, 실제 이러한 정보들은 손쉽게 추가 및 변경이 가능하다. 사용자는 각 정보나 서비스에 대한 상태를 확인하고 싶을 때 show 명령을 통해 보다 효율적으로 소프트웨어 개발을 수행할 수 있다.

표 2. Cross Platform CLI(show)에 필요한 기능 목록
Table 2. Commands included in the show function required in the Cross Platform CLI

명령어	설명
show fw	프레임워크 설정 정보 전시
show nodelist	전체 node list 전시
show node 번호	지정된 node 정보 전시
show ipclist	전체 ipc list 전시
show ipc 번호	지정된 ipc 상세 정보 전시
show tasklist	전체 Task list 정보 전시
show task 번호	지정된 Task 상세정보 전시
show timerlist	전체 Timer list 정보 전시
show timer 번호	지정된 Timer 상세 정보 전시
show msglist	전체 Msg list 정보 전시
show msg msgId	지정된 Msg 상세 정보 전시
show routinelist	전체 routine list 정보 전시
show routine 번호	지정된 routine 상세 정보 전시
show resourcelist	전체 resource list 정보 전시
show resource 번호	지정된 resource 상세정보 전시
show logconf	전체 log 설정 정보 전시
show loggrouplist	전체 log group list 정보 전시

2.3.3 process CLI command (change)

Cross Platform CLI의 입력가능상태에서 modify 명령어를 이용해 시스템이 제공하는 데이터를 변경 및 조작할 수 있는 기능이 필요하다. 기본적으로 내장형 소프트웨어가 제공하는 서비스들의 데이터를 변경 및 조작을 할 수 있으며, 본 논문에서는 NSFW의 데이터를 예로 들어 설명한다. CLI 서비스에서 변경 및 조작할 수 있는 정보들은 각종 통신들의 on/off, 태스크(task)의 start/stop/suspend/resume, 타이머(timer)의 stop, 메시지(message)의 on/off, 루틴(routine)의 on/off, 로그(log)의 on/off, 레벨 변경 정보까지 다양한 변경 및 조작 기능을 제공한다.

예를 들어 개발자의 명령(change ipc 0 use off)으로 특정 통신 인터페이스를 사용중지(off)할 수 있다. 그리고 해당 통신 인터페이스를 사용중지(off) 한 이후에 앞 절에서 언급한 현재 상태를 보여주는 명령(show ipc 0)를 통해서 실제 Use Flag가 off로 설정되었음을 확인할 수 있다. 내장형 소프트웨어 개발 및 시험을 할 때 구동 하드웨어 환경에서 키보드나 마우스와 같은 입력장치가 없어 실제 연결되어 통신 중인 IPC(Inter Process Communication)의 정보들을 제어하기 어려울 때 이 기능을 사용하면 편리하게 해당 정보를 변경할 수 있다.

표 3은 Cross Platform CLI에서 필요한 change 기능에 포함되는 명령어들을 NSFW의 데이터를 기준으로 제어(change)하는 기능으로 나열하였다. 각 서비스에 대한 제어가 필요할 때 사용하면 보다 효율적으로 소프트웨어 개발을 수행할 수 있다.

표 3. Cross Platform CLI(show)에 필요한 기능 목록
Table 3. Commands included in the change function required in the Cross Platform CLI

명령어	설명
change ipc 번호 use on/off	통신 사용 여부 on/off
change task 번호 state start/stop/suspend/resume	태스크 상태 변경(start/stop/suspend/resume)
change timer 번호 state stop	타이머 상태 변경(start -> stop)
change msg 번호 use on/off	메시지 사용 여부 on/off
change routine 번호 use on/off	루틴 사용 여부 on/off
change logconf group add/del 로그그룹이름	로그 그룹 사용 여부 add/del 수행
change logconf group 로그그룹이름 포트번호 level	출력 로그 레벨 변경
change logport 로그그룹이름 포트번호 use on/off	로그포트 사용 여부 on/off

2.3.4 process CLI command (restart)

Cross Platform CLI에 접속 및 인증이 성공한 상태인 입력가능 상태에서, 재시작(restart) 명령어를 이용해 현재 동작 중인 소프트웨어들을 모두 종료하고 다시 시작하는 기능이 필요하다. 본 논문에서는 NSFW를 예로 들어 설명한다. NSFW가 제공하는 restart 명령을 사용하게 되면 제공하는 서비스들을 모두 종료하고 다시 시작할 수 있는 기능이다. 새로운 설정을 적용하여 시험할 때에도 유용하게 쓰일 수 있으며, 특히 동적 모듈(Dynamic Module) 서비스와 연계하여 다양한 알고리즘을 시험하기 위해 원격으로 재시작할 수 있는 기능을 편리하게 사용할 수 있다. 실제 방위 산업에서 적용된 사례를 예로 들면, 전투기에 탑재된 Cross Platform CLI서비스의 restart를 이용하여 다양한 전자전 공격 알고리즘을 시험해 볼 수 있었다. 군 운용 시험(Operating Test)에서 전투기에 다양한 대체가능한 알고리즘을 미리 탑재하고 나서 실제 상황에서 다양한 알고리즘을 테스트하기 위해서 전투기를 매번 착륙하고 개발자가 알고리즘을 교체하고 다시 이륙하여 시험하는 방식은 현실적으로 매우 힘들다. 따라서 본 논문에서 고안한 방법을 활용하여 군 운용시험에서 전투기는 계속 기동을 하고 있으면서 개발자가 원격으로 CLI를 통해서 접속하여 동적 모듈을 교체하고 restart 명령을 통해 해당 소프트웨어를 재가동하여 한 번의 비행으로 다양한 알고리즘을 시험해 볼 수 있었다.

2.3.5 process CLI command (monitor)

Cross Platform CLI에 접속 및 인증이 성공한 상태인 입력가능 상태에서, 모니터(monitor) 명령어를 이용해 시스템이 제공하는 데이터를 실시간으로 확인할 수 있는 기능이 필요하다. 기본적으로 내장형 소프트웨어가 제공하는 서비스들의 중요한 데이터를 실시간으로 확인할 수 있으며, 본 논문에서는 NSFW의 데이터를

예로 들어 설명한다. 프레임워크와 응용 소프트웨어의 중요한 상태인 노드(Node)의 활성화(Live) 여부, 타이머(Timer)의 활성화 여부, 통신(IPC) 활성화 여부 및 송수신양, 태스크(Task)의 활성화 여부의 변화를 실시간으로 전시하는 기능이다. 마치 리눅스에서 실시간으로 cpu 사용을 측정하는데 자주쓰이는 top 명령어와 유사한 기능으로 시스템 관제시에 유용하게 사용할 수 있다. 또한 실시간 모니터링을 종료하기 위한 기능이 필요하며, NSFW에서는 종료하는 방법으로 명령어(q)를 입력하는 방식으로 설계하였다.

2.3.6 process CLI command (prt)

Cross Platform CLI에 접속 및 인증이 성공한 상태인 입력가능 상태에서, 프린트(prt) 명령어를 이용해 특정 로그의 출력을 CLI를 통해 전시하는 기능이 필요하다. 기본적으로 내장형 소프트웨어가 제공하는 로그들의 설정된 로그 출력방법이 아닌 CLI로 확인할 수 있으며, 본 논문에서는 NSFW의 데이터를 예로 들어 설명한다. 타겟(Target) 하드웨어에 디스플레이가 없는 상황에서 원하는 로그를 원격으로 보고 싶을 때 사용할 수 있다. 예를 들면 원격으로 출력하고자하는 로그그룹명(예: prt frwout)을 입력해주면 해당 그룹의 로그(예: hello world)가 CLI를 통해서 전시가 된다. 내장형 시스템에 복잡해질수록 로그들의 종류도 다양해지므로 각 로그들의 효과적인 활용을 위해서 구분하여 처리가 필요한데 이러한 기능을 통해서 특정 로그에 대해서는 CLI를 통해서 출력할 수 있게 되어 보다 다양한 상황에서 효과적으로 소프트웨어를 제어할 수 있다.

2.3.7 process CLI command (etc)

추가로 Cross Platform CLI에 접속 및 인증이 성공한 상태인 입력가능 상태에서 편리하게 사용할 수 있는 기타 명령들이 개발에 도움을 줄 수 있다. CLI에서 실제 사용할 수 있는 각 기능들을 간단하게 전시하는 명령어(ls)와 각 기능의 상세한 설명을 전시하는 명령어(help), 그리고 화면에 출력된 내용을 지울 수 있는 명령어(clear), 마지막으로 CLI를 종료할 수 있는 명령어(q)가 있다. 이러한 기타 기능들은 각 시스템에 맞게 개발자가 손쉽게 추가하여 각 시스템에서 콜백(Callback) 형태로 선언해주면 해당 기능들을 연결하여 사용할 수 있도록 설계하는 것이 필요하다.

4. 결론

내장형 소프트웨어의 규모와 복잡도가 크게 증가하면서 개발의 생산성에 대한 필요성이 강조되고 있다. 이러한 필요성을 만족시키기 위해서는 내장형 소프트웨어의 보다 효율적인 개발환경이 필요하다. 특히 국방 사업들의 특성상 다품종 소량생산이 많고 다양한 운영체제와 다양한 하드웨어 환경이 많다. 이러한 환경에서 크로스 플랫폼을 이용하여 공통 명령을 처리할 수 있는 Cross Platform CLI서비스는 매우 효과적일 수 있다. 이에 내장형 소프트웨어에서 개발 및 시험에 유용한 기능들을 분석해서 몇가지 기능들을 개발하여 해당 소프트웨어를 LINUX에서 자체 개발한 기반 SW 프레임워크(NSFW)에서 서비스로 탑재하였다. 그리고 본 논문에서는 해당 Cross Platform CLI 서비스에 대하여 기술하였다.

본 연구를 통해 고안한 기반 소프트웨어 프레임워크 Cross Platform CLI 서비스를 활용한 내장형 소프트웨어의 효율적인 개발 방안은 개발 및 시험에 편리함을 주어, 개발 및 시험기간을 단축하는 효과를 기대할 수 있으며 나아가 내장형 소프트웨어 개발 생산성을 높일 수 있을 것이다.

그리고 실제 기반 소프트웨어 프레임워크 Cross Platform CLI 서비스를 활용하여 개발된 국방 사업들(000 지대공유도무기 사업, 항공용 000 사업, 000 통합소나체계 등)을 사례로 보면 개발시 디버깅할 때 정적분석에서는 찾지 못했던 버그들을 찾는 데 도움이 되었다고 하고, 특히 통합 시험과 운용 시험 중 개발 장비에 직접 접근이 어려울 때 Cross Platform CLI 서비스를 통해 원격 접속이 가능해서 장비에 탑재된 소프트웨어의 상태를 실시간으로 확인할 수 있고 제어할 수 있어서 시험 성공에 큰 도움이 되었다고 한다.

마지막으로 현재는 개발 및 시험 시에만 Cross Platform CLI 서비스를 사용하는 것으로 가정하여 보안이 취약한 텔넷(Telnet)을 통해서 명령어와 결과를 송수신하는데, 추후에는 보안이 강화된 SSH(Secure Shell)와 같은 프로토콜을 사용하는 것을 계획하고 있다. 또한 기존처럼 암호를 사용하는 사용자 인증 방식은 자동화된 스크립트를 사용한 brute-force 공격 등과 같은 다양한 사이버 공격에 취약할 수밖에 없기 때문에 추후에는 보안이 강화된 SSH key 적용 방식으로 개선이 필요하다^{7,8)}. 그리고 현재 개발된 기본 기능을 발전시켜 보다 효율적으로 개발 및 시험을 수행할 수 있도록 추가 기능을 개발하는 것이 향후 과제로 남아있다.

References

- [1] M. R. S. Lande, M. S. Ali, "Synchronization in Embedded Real-Time Operating Systems," International Journal of Advance Engineering and Research Development (IJAERD), vol. 1, no. 6, 2014.
- [2] T. Yu, "Testing Embedded System Applications," Computer Science and Engineering: Theses, Dissertations, and Student Research, 2010.
- [3] C. Ebert and C. Jones, "Embedded Software: Facts, Figures, and Future," IEEE Computer Society, vol. 42, no. 4, pp. 42-52, Apr. 2009.
- [4] P. Belt, J. Harkonen, M. Mottonen, P. Kess and H. Haapasalo, "Improving the Efficiency of Verification and Validation," International Journal of Services and Standards, Vol. 4, No. 2, pp. 150-166, 2008.
- [5] T. Henzinger and J. Sifakis, "The discipline of embedded systems design," IEEE Trans. Computer, Vol. 40, No. 10, pp. 32-40, 2007.
- [6] D. Lee, H. Song, J. Jin, K. Cho, "Efficient Test Method for Embedded Software Using Next Software Framework Test Service," Journal of KIISE, Vol. 47, No. 9, pp. 804-813, 2020.
- [7] J. Lee, S. Kim, J. Woo and C. Park, "Analysis and response of SSH brute force attacks in multiuser computing environment," Korea Information Processing Society, Transactions on Computer and Communication Systems, vol. 4, no. 6, pp. 205-212, 2015.
- [8] J. Park, J. Kim, B. B. Gupta, N. Park, "Network Log-Based SSH Brute-Force Attack Detection Model," CMC-COMPUTERS MATERIALS & CONTINUA, Vol. 68, no. 1, pp. 887-901, 2021.