(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2023-0072127
(43) 공개일자 2023년05월24일

(51) 국제특허분류(Int. Cl.)
G06F 21/52 (2013.01) *G06F 11/07* (2006.01)
G06F 21/55 (2013.01) *H04L 9/40* (2022.01)
(52) CPC특허분류
G06F 21/52 (2013.01)
G06F 11/0757 (2013.01)
(21) 출원번호 10-2021-0158537
(22) 출원일자 2021년11월17일
심사청구일자 없음

(71) 출원인
현대모비스 주식회사
서울특별시 강남구 테헤란로 203 (역삼동)
(72) 발명자
이재영
경기도 용인시 기흥구 마북로240번길 17-2
(74) 대리인
특허법인아주

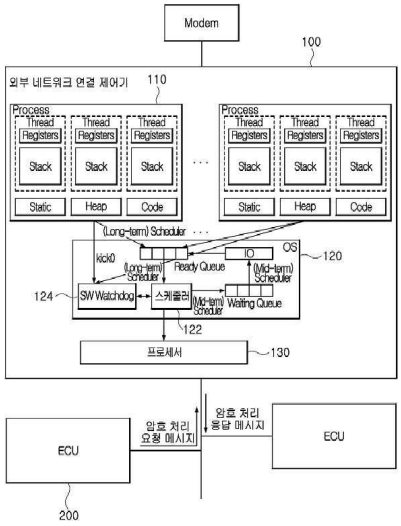
전체 청구항 수 : 총 18 항

(54) 발명의 명칭 **차량 네트워크 침입 탐지 시스템 및 방법**

(57) 요 약

차량 네트워크 침입 탐지 시스템 및 방법이 개시된다. 본 발명의 일 측면에 따른 차량 네트워크 침입 탐지 시스템은, 차량 외부 네트워크와 연결되며, SW 와치독(Software watchdog)을 사용하여 스레드(thread)의 이상 유무를 검출하는 외부 네트워크 연결 제어기, 내부 네트워크 기반으로 암호 처리 요청 메시지를 상기 SW 와치독에 전송하여 밸리데이트(validate) 함수가 수행되도록 하고, 상기 암호 처리 요청 메시지에 대한 응답 메시지를 통해 외부 침입을 판단하는 복수의 ECU를 포함한다.

대 표 도 - 도1



(52) CPC특허분류
G06F 21/554 (2013.01)
H04L 63/1408 (2013.01)
H04L 67/12 (2022.05)

명 세 서

청구범위

청구항 1

차량 외부 네트워크와 연결되며, SW 와치독(Software watchdog)을 사용하여 스레드(thread)의 이상 유무를 검출하는 외부 네트워크 연결 제어기; 및

내부 네트워크 기반으로 암호 처리 요청 메시지를 상기 SW 와치독에 전송하여 밸리데이트(validate) 함수가 수행되도록 하고, 상기 암호 처리 요청 메시지에 대한 응답 메시지를 통해 외부 침입을 판단하는 복수의 ECU(Electronic Control Unit)

를 포함하는 차량 네트워크 침입 탐지 시스템.

청구항 2

제1항에 있어서,

상기 외부 네트워크 연결 제어기는,

프로세서;

복수의 스레드(thread)로 구성되며, 스레드별 SW 와치독 필요 여부, SW 와치독이 필요한 스레드의 경우 타임아웃(timeout), 및 비정상 액션이 설정되는 복수의 프로세스; 및

상기 복수의 프로세스 및 각 프로세스내 스레드들을 정책에 따라 상기 프로세서를 점유하여 동작시키고, 상기 스레드들의 이상 유무를 검출하는 OS(Operating System) 모듈을 포함하는 것을 특징으로 하는 차량 네트워크 침입 탐지 시스템.

청구항 3

제2항에 있어서,

상기 프로세스는,

응용프로그램 시작 시, 병렬 처리를 위한 복수의 스레드를 생성하고, 각 스레드에 반복 구문을 1회 수행하는데 걸리는 최대 시간을 타임아웃으로 설정하며, 각 스레드가 타임아웃 내에 킥 신호를 상기 SW 와치독으로 전송함으로써, 해당 스레드가 정상임을 알리는 것을 특징으로 하는 차량 네트워크 침입 탐지 시스템.

청구항 4

제3항에 있어서,

상기 프로세스는,

상기 스레드 생성 시, SW 와치독 필요 여부를 판단하고, SW 와치독이 필요한 스레드인 경우 전체 스레드 수를 '1' 증가시키며, 각 스레드에 설정된 타임아웃 및 비정상 액션을 저장하고, 해당 스레드에 메모리 공간을 할당하는 것을 특징으로 하는 차량 네트워크 침입 탐지 시스템.

청구항 5

제2항에 있어서,

상기 OS 모듈은,

응용프로그램의 실행 시, TDM(time division multiplexing) 방식을 이용하여 각 스레드를 동작시키고, 각 스레드의 수행 시작 시간을 저장하여 상기 프로세서를 점유한 시간을 카운팅하는 스케줄러; 및

상기 스케줄러에서 카운팅되는 각 스레드의 수행 시간을 누적하고, 현재 수행 중인 스레드에 설정된 타임아웃 내에 킥(kick) 신호 수신 여부에 따라 해당 스레드의 침입을 감지하는 상기 SW 와치독을 포함하는 것을 특징으로 하는 차량 네트워크 침입 탐지 시스템.

청구항 6

제5항에 있어서,

상기 SW 와치독은,

상기 현재 수행 중인 스레드로부터 타임아웃 내에 킥 신호를 수신한 경우, 상기 현재 수행 중인 스레드의 누적 수행 시간을 초기화함으로써, 상기 현재 수행 중인 스레드에는 외부 침입이 발생하지 않는 것으로 판단하고,

상기 타임아웃 내에 킥 신호를 수신하지 않은 경우, 상기 현재 수행 중인 스레드에 외부 침입이 발생한 것으로 판단하는 것을 특징으로 하는 차량 네트워크 침입 탐지 시스템.

청구항 7

제5항에 있어서,

상기 SW 와치독은,

현재 수행중인 스레드, 전체 스레드 수, 비정상 액션을 수행한 스레드, 각 스레드의 타임아웃, 각 스레드의 누적 수행 시간, 및 각 스레드의 비정상 액션 중 적어도 하나를 관리하는 것을 특징으로 하는 차량 네트워크 침입 탐지 시스템.

청구항 8

제5항에 있어서,

상기 SW 와치독은,

현재 수행중인 스레드로부터 킥 신호가 수신된 경우, 현재 수행중인 스레드의 누적 수행 시간을 초기화하는 킥 함수 또는,

상기 ECU로부터 암호 처리 요청 메시지 수신 시, 모든 스레드에 대해 누적 수행 시간이 타임아웃 보다 더 큰 스레드가 있는지 확인하고, 있을 경우 등록된 비정상 액션을 수행하도록 하는 밸리데이트 함수를 수행하는 것을 특징으로 하는 차량 네트워크 침입 탐지 시스템.

청구항 9

제8항에 있어서,

상기 암호 처리 요청 메시지는, 암호 알고리즘 및 시드(seed) 값을 포함하고,

상기 SW 와치독은,

상기 암호 처리 요청 메시지 수신 시, 상기 밸리데이트 함수를 실행하고, 상기 밸리데이트 함수의 실행 결과 모든 스레드가 정상인 경우, 상기 암호 알고리즘을 동작하며, 상기 시드 값에 대한 처리 결과를 상기 응답 메시지에 포함시켜 상기 ECU로 전송하고,

상기 밸리데이트 함수의 실행 결과, 적어도 하나의 스레드가 비정상인 경우, 상기 응답 메시지를 상기 ECU로 전

송하지 않은 것을 특징으로 하는 차량 네트워크 침입 탐지 시스템.

청구항 10

제9항에 있어서,

상기 ECU는,

상기 SW 와치독으로부터 기 설정된 일정 시간 내에 상기 응답 메시지가 수신되지 않은 경우 외부 침입이라고 판단하는 것을 특징으로 하는 차량 네트워크 침입 탐지 시스템.

청구항 11

스케줄러가, 응용프로그램의 실행 시, TDM(time division multiplexing) 방식을 이용하여 각 스레드를 동작시키고, 각 스레드의 수행 시작 시간을 저장하여 프로세서를 점유한 시간을 카운팅하는 단계; 및

SW 와치독이, 상기 스케줄러에서 카운팅되는 각 스레드의 수행 시간을 누적하고, 현재 수행 중인 스레드에 설정된 타임아웃 내에 킥(kick) 신호 수신 여부에 따라 해당 스레드의 외부 침입을 감지하는 단계

를 포함하는 차량 네트워크 침입 탐지 방법.

청구항 12

제11항에 있어서,

상기 응용프로그램은,

복수의 스레드(thread)로 구성되며, 스레드별 SW 와치독 필요 여부, SW 와치독이 필요한 스레드의 경우 타임아웃(timeout), 및 비정상 액션이 설정되는 것을 특징으로 하는 차량 네트워크 침입 탐지 방법.

청구항 13

제11항에 있어서,

상기 응용프로그램의 실행 시,

프로세스가 병렬 처리를 위한 복수의 스레드를 생성하고, 각 스레드에 반복 구문을 1회 수행하는데 걸리는 최대 시간을 타임아웃으로 설정하며,

상기 생성된 각 스레드는 해당 타임아웃 내에 킥 신호를 상기 SW 와치독으로 전송함으로써, 해당 스레드가 정상임을 알리는 것을 특징으로 하는 차량 네트워크 침입 탐지 방법.

청구항 14

제13항에 있어서,

상기 복수의 스레드를 생성할 때,

상기 프로세스는, SW 와치독 필요 여부를 판단하고, SW 와치독이 필요한 스레드인 경우 전체 스레드 수를 '1' 증가시키며, 각 스레드에 설정된 타임아웃 및 비정상 액션을 저장하고, 해당 스레드에 메모리 공간을 할당하는 것을 특징으로 하는 차량 네트워크 침입 탐지 방법.

청구항 15

제11항에 있어서,

상기 해당 스레드의 외부 침입을 감지하는 단계에서,

상기 SW 와치독은, 상기 현재 수행 중인 스레드로부터 타임아웃 내에 킥 신호가 수신된 경우, 상기 현재 수행 중인 스레드의 누적 수행 시간을 초기화함으로써, 상기 현재 수행 중인 스레드에는 외부 침입이 발생하지 않는 것으로 판단하고,

상기 타임아웃 내에 킥 신호가 수신되지 않은 경우, 상기 현재 수행 중인 스레드에 외부 침입이 발생한 것으로 판단하는 것을 특징으로 하는 차량 네트워크 침입 탐지 방법.

청구항 16

제11항에 있어서,

상기 해당 스레드의 외부 침입을 감지하는 단계 이후,

ECU로부터 내부 네트워크 기반으로 암호 처리 요청 메시지 수신 시, 상기 SW 와치독이 밸리데이트(validate) 함수를 수행함으로써, 차량 네트워크의 외부 침입을 감지하는 단계를 더 포함하는 것을 특징으로 하는 차량 네트워크 침입 탐지 방법.

청구항 17

제16항에 있어서,

상기 차량 네트워크의 외부 침입을 감지하는 단계에서,

상기 암호 처리 요청 메시지는, 암호 알고리즘 및 시드(seed) 값을 포함하고,

상기 SW 와치독은, 상기 암호 처리 요청 메시지 수신 시, 상기 밸리데이트 함수를 실행하고, 상기 밸리데이트 함수의 실행 결과 모든 스레드가 정상인 경우, 상기 암호 알고리즘을 동작하며, 상기 시드 값에 대한 처리 결과를 상기 응답 메시지에 포함시켜 상기 ECU로 전송하고,

상기 밸리데이트 함수의 실행 결과, 적어도 하나의 스레드가 비정상인 경우, 상기 응답 메시지를 상기 ECU로 전송하지 않은 것을 특징으로 하는 차량 네트워크 침입 탐지 방법.

청구항 18

제17항에 있어서,

상기 ECU는, 상기 SW 와치독으로부터 기 설정된 일정 시간 내에 상기 응답 메시지가 수신되지 않은 경우 외부 침입이라고 판단하는 것을 특징으로 하는 차량 네트워크 침입 탐지 방법.

발명의 설명

기 술 분 야

[0001] 본 발명은 차량 네트워크 침입 탐지 시스템 및 방법에 관한 것으로, 보다 상세하게는 제어기 동작 상태에서 SW 와치독(Software watchdog)을 사용하여 외부 침입을 감지할 수 있도록 하는 차량 네트워크 침입 탐지 시스템 및 방법에 관한 것이다.

배 경 기 술

[0002] 차량은 다수의 제조 업체에서 제작된 부품들이 조립되어 만들어지기 때문에 다양한 SW 개발 방법으로 만들어진 기능들이 호환되도록 제어기 사이에 약한 연결 관계를 갖는다. 또한 실시간 제어를 위하여 제어기 사이에 전송하는 데이터를 고도로 암호화하기 어렵기 때문에 쉽게 해킹의 위험에 노출될 수 있다. 더욱이 차량의 기능이 고도화 되면서 무선 통신망을 통하여 외부 네트워크에 접속함에 따라 사이버 보안의 중요성은 더욱 더 커지고 있

다.

[0003] 일반적으로 차량용 제어기는 외부 침입에 대하여 강건성을 확보하기 위하여 HSM(Hardware Security Module)이 적용된 MCU(Micro Controller Unit)/AP(Application Processor)등을 사용하여 secure boot나 secure flash 기능을 구현한다. Secure boot는 firmware 영역을 SHA-256 등의 방법을 사용하여 검사함으로써 변경이 없을 때에만 boot를 시작하며, secure flash는 인증된 장비나 사용자만 flash write가 가능하도록 함으로써 침입자에 의하여 비휘발성 메모리(non-volatile memory) 영역 변경을 할 수 없도록 한다.

[0004] 외부 침입자의 공격에 의하여 제어기 동작 제어권을 빼앗겼을 경우, 침입자의 SW는 휘발성 메모리(volatile memory) 영역에 존재하게 된다. 따라서 제어기에 전원이 유지되는 상황에서는 침입자에 의하여 차량 내부 망에 잘못된 데이터가 전송되어 차량이 악의적으로 제어될 수 있다. 제어권을 빼앗긴 제어기가 전송 주기가 짧은 메시지를 전송하여 차량 제어를 정상적으로 수행하지 못하게 하는 것은 약속된 주기가 아닌 메시지를 검출만 하면 되므로 검출이 쉽다.

[0005] 하지만 외부 네트워크와 직접 연결되어 있는 CCU(Central Communication Unit)의 제어권을 빼앗겼을 경우 게이트웨이(gateway) 과정에서 이상 데이터를 전송하면 되므로 차량 내부 다른 망에 연결된 제어기에 잘못된 정보를 송신하여 차량이 이상 제어될 수 있다. 특히 이더넷으로 연결된 제어기는 스위치와 1대 1로 연결되어 있으므로, 스위치가 장착된 CCU의 제어권을 빼앗겼을 경우 통신을 통한 입력 정보를 믿을 수 없게 된다.

[0006] 이에, 차량이 외부 네트워크에 연결되면서 악의적인 외부 침입을 방지하기 위하여 사이버 보안이 강조되고 있다. 하지만 차량에 적용된 사이버 보안 대응 방법은 비휘발성 메모리(non-volatile memory)나 이상 주기를 갖는 신호 탐지에 국한되어 있으므로 외부 침입자의 공격에 의하여 휘발성 메모리(volatile memory) 영역 침입 감지에는 취약하다. 이로 인해 제어기의 동작 중에 외부 침입자에 의하여 차량 내부 망에 잘못된 데이터가 전송되어 차량이 악의적으로 제어되는 문제가 발생한다.

[0007] 본 발명의 배경기술은 대한민국 공개특허공보 제10-2021-0075386호(2021.06.23. 공개)에 개시되어 있다.

발명의 내용

해결하려는 과제

[0008] 본 발명은 상기와 같은 문제점들을 개선하기 위하여 안출된 것으로, 본 발명의 목적은 제어기 동작 상태에서 외부 침입을 감지할 수 있도록 하는 차량 네트워크 침입 탐지 시스템 및 방법을 제공하는 것이다.

[0009] 본 발명이 해결하고자 하는 과제는 이상에서 언급한 과제(들)로 제한되지 않으며, 언급되지 않은 또 다른 과제(들)은 아래의 기재로부터 당업자에게 명확하게 이해될 수 있을 것이다.

과제의 해결 수단

[0010] 본 발명의 일 측면에 따른 차량 네트워크 침입 탐지 시스템은, 차량 외부 네트워크와 연결되며, SW 와치독(Software watchdog)을 사용하여 스레드(thread)의 이상 유무를 검출하는 외부 네트워크 연결 제어기, 내부 네트워크 기반으로 암호 처리 요청 메시지를 상기 SW 와치독에 전송하여 밸리데이트(validate) 함수가 수행되도록 하고, 상기 암호 처리 요청 메시지에 대한 응답 메시지를 통해 외부 침입을 판단하는 복수의 ECU를 포함한다.

[0011] 본 발명에서 상기 외부 네트워크 연결 제어기는, 프로세서, 복수의 스레드(thread)로 구성되며, 스레드별 SW 와치독 필요 여부, SW 와치독이 필요한 스레드의 경우 타임아웃(timeout), 및 비정상 액션이 설정되는 복수의 프로세스, 상기 복수의 프로세스 및 각 프로세스내 스레드들을 정책에 따라 상기 프로세서를 점유하여 동작시키고, 상기 스레드들의 이상 유무를 검출하는 OS(Operating System) 모듈을 포함할 수 있다.

[0012] 본 발명에서 상기 프로세스는, 응용프로그램 시작 시, 병렬 처리를 위한 복수의 스레드를 생성하고, 각 스레드에 반복 구문을 1회 수행하는데 걸리는 최대 시간을 타임아웃으로 설정하며, 각 스레드가 타임아웃 내에 킥 신호를 상기 SW 와치독으로 전송함으로써, 해당 스레드가 정상임을 알릴 수 있다.

[0013] 본 발명에서 상기 프로세스는, 상기 스레드 생성 시, SW 와치독 필요 여부를 판단하고, SW 와치독이 필요한 스레드인 경우 전체 스레드 수를 '1' 증가시키며, 각 스레드에 설정된 타임아웃 및 비정상 액션을 저장하고, 해당 스레드에 메모리 공간을 할당할 수 있다.

[0014] 본 발명에서 상기 OS 모듈은, 응용프로그램의 실행 시, TDM(time division multiplexing) 방식을 이용하여 각

스레드를 동작시키고, 각 스레드의 수행 시작 시간을 저장하여 상기 프로세서를 점유한 시간을 카운팅하는 스케줄러, 및 상기 스케줄러에서 카운팅되는 각 스레드의 수행 시간을 누적하고, 현재 수행 중인 스레드에 설정된 타임아웃 내에 킥(kick) 신호 수신 여부에 따라 해당 스레드의 침입을 감지하는 상기 SW 와치독을 포함할 수 있다.

[0015] 본 발명에서 상기 SW 와치독은, 상기 현재 수행 중인 스레드로부터 타임아웃 내에 킥 신호를 수신한 경우, 상기 현재 수행 중인 스레드의 누적 수행 시간을 초기화함으로써, 상기 현재 수행 중인 스레드에는 외부 침입이 발생하지 않는 것으로 판단하고, 상기 타임아웃 내에 킥 신호를 수신하지 않은 경우, 상기 현재 수행 중인 스레드에 외부 침입이 발생한 것으로 판단할 수 있다.

[0016] 본 발명에서 상기 SW 와치독은, 현재 수행중인 스레드, 전체 스레드 수, 비정상 액션을 수행한 스레드, 각 스레드의 타임아웃, 각 스레드의 누적 수행 시간, 및 각 스레드의 비정상 액션 중 적어도 하나를 관리할 수 있다.

[0017] 본 발명에서 상기 SW 와치독은, 현재 수행중인 스레드로부터 킥 신호가 수신된 경우, 현재 수행중인 스레드의 누적 수행 시간을 초기화하는 킥 함수 또는, 상기 ECU로부터 암호 처리 요청 메시지 수신 시, 모든 스레드에 대해 누적 수행 시간이 타임아웃 보다 더 큰 스레드가 있는지 확인하고, 있을 경우 등록된 비정상 액션을 수행하도록 하는 밸리데이트 함수를 수행할 수 있다.

[0018] 본 발명에서 상기 암호 처리 요청 메시지는, 암호 알고리즘 및 시드(seed) 값을 포함하고, 상기 SW 와치독은, 상기 암호 처리 요청 메시지 수신 시, 상기 밸리데이트 함수를 실행하고, 상기 밸리데이트 함수의 실행 결과 모든 스레드가 정상인 경우, 상기 암호 알고리즘을 동작하며, 상기 시드 값에 대한 처리 결과를 상기 응답 메시지에 포함시켜 상기 ECU로 전송하고, 상기 밸리데이트 함수의 실행 결과, 적어도 하나의 스레드가 비정상인 경우, 상기 응답 메시지를 상기 ECU로 전송하지 않을 수 있다.

[0019] 본 발명에서 상기 ECU는, 상기 SW 와치독으로부터 기 설정된 일정 시간 내에 상기 응답 메시지가 수신되지 않은 경우 외부 침입이라고 판단할 수 있다.

[0020] 본 발명의 일 측면에 따른 차량 네트워크 침입 탐지 방법은, 스케줄러가, 응용프로그램의 실행 시, TDM(time division multiplexing) 방식을 이용하여 각 스레드를 동작시키고, 각 스레드의 수행 시작 시간을 저장하여 프로세서를 점유한 시간을 카운팅하는 단계, 및 SW 와치독이, 상기 스케줄러에서 카운팅되는 각 스레드의 수행 시간을 누적하고, 현재 수행 중인 스레드에 설정된 타임아웃 내에 킥(kick) 신호 수신 여부에 따라 해당 스레드의 외부 침입을 감지하는 단계를 포함한다.

[0021] 본 발명에서 상기 응용프로그램은, 복수의 스레드(thread)로 구성되며, 스레드별 SW 와치독 필요 여부, SW 와치독이 필요한 스레드의 경우 타임아웃(timeout), 및 비정상 액션이 설정될 수 있다.

[0022] 본 발명에서 상기 응용프로그램의 실행 시, 프로세스가 병렬 처리를 위한 복수의 스레드를 생성하고, 각 스레드에 반복 구문을 1회 수행하는데 걸리는 최대 시간을 타임아웃으로 설정하며, 상기 생성된 각 스레드는 해당 타임아웃 내에 킥 신호를 상기 SW 와치독으로 전송함으로써, 해당 스레드가 정상임을 알릴 수 있다.

[0023] 본 발명에서 상기 복수의 스레드를 생성할 때, 상기 프로세스는, SW 와치독 필요 여부를 판단하고, SW 와치독이 필요한 스레드인 경우 전체 스레드 수를 '1' 증가시키며, 각 스레드에 설정된 타임아웃 및 비정상 액션을 저장하고, 해당 스레드에 메모리 공간을 할당할 수 있다.

[0024] 본 발명은 상기 해당 스레드의 외부 침입을 감지하는 단계에서, 상기 SW 와치독은, 상기 현재 수행 중인 스레드로부터 타임아웃 내에 킥 신호가 수신된 경우, 상기 현재 수행 중인 스레드의 누적 수행 시간을 초기화함으로써, 상기 현재 수행 중인 스레드에는 외부 침입이 발생하지 않는 것으로 판단하고, 상기 타임아웃 내에 킥 신호가 수신되지 않은 경우, 상기 현재 수행 중인 스레드에 외부 침입이 발생한 것으로 판단할 수 있다.

[0025] 본 발명은 상기 해당 스레드의 외부 침입을 감지하는 단계 이후, ECU로부터 내부 네트워크 기반으로 암호 처리 요청 메시지 수신 시, 상기 SW 와치독이 밸리데이트(validate) 함수를 수행함으로써, 차량 네트워크의 외부 침입을 감지하는 단계를 더 포함할 수 있다.

[0026] 본 발명은 상기 차량 네트워크의 외부 침입을 감지하는 단계에서, 상기 암호 처리 요청 메시지는, 암호 알고리즘 및 시드(seed) 값을 포함하고, 상기 SW 와치독은, 상기 암호 처리 요청 메시지 수신 시, 상기 밸리데이트 함수를 실행하고, 상기 밸리데이트 함수의 실행 결과 모든 스레드가 정상인 경우, 상기 암호 알고리즘을 동작하며, 상기 시드 값에 대한 처리 결과를 상기 응답 메시지에 포함시켜 상기 ECU로 전송하고, 상기 밸리데이

트 함수의 실행 결과, 적어도 하나의 스레드가 비정상인 경우, 상기 응답 메시지를 상기 ECU로 전송하지 않을 수 있다.

[0027] 본 발명에서 상기 ECU는, 상기 SW 와치독으로부터 기 설정된 일정 시간 내에 상기 응답 메시지가 수신되지 않은 경우 외부 침입이라고 판단할 수 있다.

발명의 효과

[0028] 본 발명의 일 측면에 따른 차량 네트워크 침입 탐지 시스템 및 방법은, 제어기 동작 상태에서 SW 와치독을 사용하여 스레드 이상 유무를 확인함으로써, 외부 침입을 감지할 수 있다.

[0029] 본 발명의 일 측면에 따른 차량 네트워크 침입 탐지 시스템 및 방법은, 제어기 내부 SW에 침입이 있는지 SW 와치독을 사용하여 모니터링하므로, 외부 침입 초기부터 감지하여 이상 제어를 방지할 수 있고, 사이버 보안 위협에 보다 강건하다.

[0030] 본 발명의 일 측면에 따른 차량 네트워크 침입 탐지 시스템 및 방법은, 외부 침입자의 SW가 휘발성 메모리(volatile memory)에 상주하는 경우도 감지 가능하므로 외부 침입에 강건한 차량 네트워크를 구성할 수 있다.

[0031] 본 발명의 일 측면에 따른 차량 네트워크 침입 탐지 시스템 및 방법은, 외부 침입에 의하여 이상 동작한 경우뿐만 아니라 SW 버그 등에 의한 이상 동작도 모니터링 가능하므로 최근 점점 더 복잡해지는 차량용 SW 생태계에서 필수적인 고장 감지 및 해결 방법이 될 것이다.

[0032] 한편, 본 발명의 효과는 이상에서 언급한 효과들로 제한되지 않으며, 이하에서 설명할 내용으로부터 통상의 기술자에게 자명한 범위 내에서 다양한 효과들이 포함될 수 있다.

도면의 간단한 설명

[0033] 도 1은 본 발명의 일 실시예에 따른 차량 네트워크 침입 탐지 시스템을 설명하기 위한 도면이다.

도 2는 본 발명의 일 실시예에 따른 SW 와치독을 설명하기 위한 도면이다.

도 3은 본 발명의 일 실시예에 따른 SW 와치독이 킥 함수를 수행하는 방법을 설명하기 위한 타이밍도이다.

도 4는 본 발명의 일 실시예에 따른 응용프로그램의 동작을 설명하기 위한 도면이다.

도 5는 본 발명의 일 실시예에 따른 스레드 생성 방법을 설명하기 위한 도면이다.

도 6은 본 발명의 일 실시예에 따른 킥 함수를 설명하기 위한 흐름도이다.

도 7은 본 발명의 일 실시예에 따른 밸리데이트 함수를 설명하기 위한 흐름도이다.

발명을 실시하기 위한 구체적인 내용

[0034] 이하에서는 본 발명의 일 실시예에 따른 차량 네트워크 침입 탐지 시스템 및 방법을 첨부된 도면들을 참조하여 상세하게 설명한다. 이러한 과정에서 도면에 도시된 선들의 두께나 구성요소의 크기 등은 설명의 명료성과 편의상 과장되게 도시되어 있을 수 있다. 또한 후술되는 용어들은 본 발명에서의 기능을 고려하여 정의된 용어들로서, 이는 이용자, 운용자의 의도 또는 관례에 따라 달라질 수 있다. 그러므로 이러한 용어들에 대한 정의는 본 명세서 전반에 걸친 내용을 토대로 내려져야 할 것이다.

[0035] 본 명세서에서 설명된 구현은, 예컨대, 방법 또는 프로세스, 장치, 소프트웨어 프로그램, 데이터 스트림 또는 신호로 구현될 수 있다. 단일 형태의 구현의 맥락에서만 논의(예컨대, 방법으로서만 논의)되었더라도, 논의된 특징의 구현은 또한 다른 형태(예컨대, 장치 또는 프로그램)로도 구현될 수 있다. 장치는 적절한 하드웨어, 소프트웨어 및 펌웨어 등으로 구현될 수 있다. 방법은, 예컨대, 컴퓨터, 마이크로프로세서, 집적 회로 또는 프로그래밍 가능한 로직 디바이스 등을 포함하는 프로세싱 디바이스를 일반적으로 지칭하는 프로세서 등과 같은 장치에서 구현될 수 있다. 프로세서는 또한 최종-사용자 사이에 정보의 통신을 용이하게 하는 컴퓨터, 셀 폰, 휴대용/개인용 정보 단말기(personal digital assistant: "PDA") 및 다른 디바이스 등과 같은 통신 디바이스를 포함한다.

[0037] 도 1은 본 발명의 일 실시예에 따른 차량 네트워크 침입 탐지 시스템을 설명하기 위한 도면, 도 2는 본 발명의

일 실시예에 따른 SW 와치독을 설명하기 위한 도면, 도 3은 본 발명의 일 실시예에 따른 SW 와치독이 킥 함수를 수행하는 방법을 설명하기 위한 타이밍도이다.

[0038] 도 1을 참조하면, 본 발명의 일 실시예에 따른 차량 네트워크 침입 탐지 시스템은 차량 외부 네트워크와 연결된 외부 네트워크 연결 제어기(100) 및 차량 내부의 하나 이상의 ECU(Electronic Control Unit)(200)를 포함한다.

[0039] 외부 네트워크 연결 제어기(100)는 제어기(100) 동작 상태에서 SW 와치독(Software watchdog)(124)을 사용하여 스레드(thread)의 이상 유무를 확인함으로써, 외부 침입을 감지할 수 있다.

[0040] 이러한 외부 네트워크 연결 제어기(100)는 복수의 프로세스(응용프로그램)(110), OS 모듈(120) 및 프로세서(130)를 포함할 수 있다.

[0041] 프로세스(110)는 복수의 스레드(thread)로 구성되며, 스레드별 SW 와치독(124) 필요 여부, SW 와치독(124)이 필요한 스레드의 경우 타임아웃(timeout) 및 비정상 액션이 설정되어 있다. 여기서, SW 와치독(124)이 필요한 스레드는 엔진, 핸들, 및 브레이크 등 차량 제어에 필요한 스레드를 포함할 수 있다. 타임아웃은 각 스레드에 반복 구문을 1회 수행하는데 걸리는 최대 시간일 수 있다. 비정상 액션은 제어기(100) 내부에서 외부 침입이 감지된 상황에 대한 대응을 나타내며 아래 표 1과 같을 수 있다.

표 1

[0042]

Action	Note
None	별도의 처리를 진행하지 않는다.
Reset	Thread를 초기화 한다. Thread가 사용한 메모리 영역을 초기화 한다
Lock	Thread를 sheduler queue에서 제거하여 호출되지 못하도록 한다.
Reboot	시스템 전원을 초기화 한다.
Down	시스템을 power down 모드로 전환한다(전기능 미동작).

[0043] None는 해당 스레드가 권한이 없는 스레드여서 별도의 처리를 하지 않는 것을 의미한다. Reset은 외부와 통신할 권한이 있고, 실시간 처리를 요하지 않는다면 스레드와 사용한 메모리 영역을 초기화하는 것을 의미한다. Lock은 실시간 처리가 필요한 스레드이며, 다시 제어권을 빼앗기기 쉬운 스레드의 경우 스케줄러 큐(scheduler queue)에서 제거하여 호출되지 못하도록 하는 것을 의미한다. Reboot는 외부 침입에 의해 NVM(non-volatile memory)까지 영향을 줄 수 있는 권한이 많은 스레드의 경우, 시스템 전원을 초기화하는 것을 의미할 수 있다. Down은 외부 침입에 의해 NVM(non-volatile memory)까지 영향을 줄 수 있는 권한이 많은 스레드의 경우, 시스템 전원을 내려서 차량 내부 시스템까지 침입이 확산되지 않도록 하는 것을 의미할 수 있다.

[0044] 프로세스(응용프로그램)(110)는 각 스레드(thread)에서 주기적으로 킥(kick) 신호를 생성하여 OS 모듈(120)의 SW 와치독(124)에 이상 없음을 알려야 한다.

[0045] 이에, 프로세스(응용프로그램)(110)는 첫 시작에서 내부 변수 등을 초기화하고 병렬 처리를 위한 스레드를 생성한다. 각 스레드는 내부에 반복 구문이 있어, 목적하는 횟수만큼 처리한 후 자원을 해제하고 프로그램을 종료한다. 즉, 응용프로그램은 동작 시작 시, 병렬 처리를 위한 복수의 스레드를 생성하고, 각 스레드에 반복 구문을 1회 수행하는데 걸리는 최대 시간을 타임아웃으로 설정하며, 각 스레드가 타임아웃 내에 킥 신호를 SW 와치독(124)으로 전송함으로써, 해당 스레드가 정상임을 알릴 수 있도록 한다. 스레드 생성 시, 응용프로그램은 SW 와치독(124) 필요 여부를 판단하고, SW 와치독(124)이 필요한 스레드인 경우 전체 스레드 수를 '1' 증가시키며, 각 스레드에 설정된 타임아웃 및 비정상 액션을 저장하고, 저장공간을 할당하여 초기화를 수행할 수 있다. 그런 후, 응용프로그램은 해당 스레드에 스택 메모리를 할당함으로써, 스레드 생성을 완료할 수 있다.

[0046] 이처럼, 응용프로그램은 반복 구문을 1회 수행하는데 걸리는 최대 시간을 타임아웃 값으로 스레드를 생성한 후, 스레드가 반복 구문 안에서 킥 신호를 생성하도록 하면, 스레드가 외부 침입에 의하여 제어권을 빼앗겼을 경우 킥 신호가 생성되지 않으므로 스레드의 이상 유무를 알 수 있다. 킥 신호가 발생하지 않아서 스레드의 누적 수행 시간이 타임아웃 보다 커질 경우, 해당 스레드는 기 설정된 비정상 액션을 수행하게 된다.

[0047] 상술한 바와 같이 응용프로그램은 반복 구문으로 되어 있으므로, 응용프로그램 시작 시, 각 스레드는 주기적으로 킥 신호를 생성하여 해당 스레드가 정상임을 알릴 수 있다. 또한, 응용프로그램은 반복 구문이 해제되었을

때 점유 자원 해제하고 응용프로그램을 종료한다.

[0048] OS 모듈(120)은 복수의 프로세스(110) 및 각 프로세스(110)내 스레드들을 정책에 따라 프로세서(130)를 점유하여 동작시키고, 스레드들의 이상 유무를 검출할 수 있다.

[0049] 이러한 OS 모듈(120)은 스케줄러(122) 및 SW 와치독(124)을 포함할 수 있다.

[0050] 스케줄러(122)는 응용프로그램의 실행 시, TDM(time division multiplexing) 방식을 이용하여 각 스레드를 동작시키고, 각 스레드의 수행 시작 시간을 저장하여 프로세서(130)를 점유한 시간을 카운팅할 수 있다.

[0051] 스케줄러(122)는 각 스레드(thread)의 수행 시작 시간을 저장하여 프로세서(130)를 차지한(점유한) 시간을 SW 와치독(124)에 전달함으로써, SW 와치독(124)이 실제 프로세서(130)를 차지한 시간 중에 킥 신호의 발생 유무를 판단할 수 있도록 한다.

[0052] 스케줄러(122)는 다수의 프로세스(110)와 프로세스(110) 내부의 스레드를 정책에 따라 동작시킨다. 그래서 단일 연산 유닛(processor, microcontroller)가 있다고 하더라도 TDM(time division multiplexing)을 사용하여 다수의 프로세스(110)와 스레드를 동작시키므로 운전자는 동시에 다수의 기능이 병렬로 동작되는 것처럼 판단하게 된다. 프로세스(110)와 스레드의 선택은 스케줄러(122)가 수행한다. 따라서 스케줄러(122)는 동작 중인 모든 프로세스(110)와 스레드 정보 및 프로세서 점유 시간 등의 정보를 알 수 있다

[0053] SW 와치독(124)은 스케줄러(122)에서 카운팅되는 각 스레드의 수행 시간을 누적하고, 현재 수행 중인 스레드에 설정된 타임아웃 내에 킥(kick) 신호 수신 여부에 따라 해당 스레드의 침입을 감지할 수 있다.

[0054] SW 와치독(124)은, 현재 수행 중인 스레드로부터 타임아웃 내에 킥 신호를 수신한 경우, 현재 수행 중인 스레드의 누적 수행 시간을 초기화함으로써, 현재 수행 중인 스레드에는 외부 침입이 발생하지 않는 것으로 판단할 수 있다. 만약, 타임아웃 내에 킥 신호를 수신하지 않으면, SW 와치독(124)은 현재 수행 중인 스레드에 외부 침입이 발생한 것으로 판단할 수 있다.

[0055] 또한, SW 와치독(124)은 현재 수행중인 스레드, 전체 스레드 수, 비정상 액션을 한 스레드, 각 스레드의 타임아웃, 각 스레드의 누적 수행 시간, 및 각 스레드의 비정상 액션 중 적어도 하나를 관리할 수 있다. 이를 위해 SW 와치독(124)은 도 2의 (a)와 같은 구조일 수 있다. 도 2의 (a)를 참조하면, SW 와치독(124)의 내부에는 현재 수행 스레드를 저장하는 공간과 전체 스레드 수를 관리하는 공간이 있다. 또한 각 스레드가 생성될 때, SW 와치독(124)은 도 2의 (b)와 같이 누적 수행 시간을 저장할 공간을 생성하고, 타임아웃과 비정상 액션을 제공받을 수 있다. 타임아웃은 해당 스레드가 정상임을 알기 위한 최대 시간일 수 있고, 비정상 액션은 킥 신호가 발생하지 않았을 때 수행할 행동 지침일 수 있다.

[0056] 또한, SW 와치독(124)은 킥 함수, 밸리데이트(validate) 함수 및 SW 와치독(124) 생성(삭제) 중 적어도 하나를 수행할 수 있다. 여기서, 킥 함수는 현재 수행중인 스레드로부터 킥 신호가 수신된 경우, 현재 수행중인 스레드의 누적 수행 시간을 초기화하는 함수일 수 있다. 밸리데이트 함수는 ECU(200)로부터 암호 처리 요청 메시지 수신 시, 모든 스레드에 대해 누적 수행 시간이 타임아웃 보다 더 큰 스레드가 있는지 확인하고, 있을 경우 등록된 비정상 액션을 수행하도록 하는 함수일 수 있다.

[0057] SW 와치독(124)은 외부 침입으로부터 스레드의 제어권을 빼앗겼을 때, 스레드 별로 설정된 타임아웃 동안 킥 신호가 수신되지 않으면 해당 스레드에 문제가 발생한 것으로 판단할 수 있다.

[0058] 일반적인 스레드의 life time을 기준으로 특정 시간 동안 킥 신호가 발생하였는지를 판단할 경우, 도 3에 도시된 바와 같이 스케줄러(122)에 의해 선점되어 미수행되는 시간이 대다수일 수 있다. 즉, 초기에 스레드가 생성된 후 우선 순위가 높은 스레드 때문에 한 번도 수행되지 않는 deadlock 상황에서도 스레드는 킥 신호를 발생시키는 것이 정상이다. 이에, 스케줄러(122)는 각 스레드의 수행 시작 시간을 저장하여 프로세서(130)를 점유한(차지한) 시간을 SW 와치독(124)에 전달해 줌으로써 실제 프로세서(130)를 차지한 시간 중에 킥 신호의 발생 유무를 판단할 수 있도록 한다.

[0059] SW 와치독(124)은 각 스레드의 수행 시간을 누적하고, 현재 수행 중인 스레드에서 킥 신호가 수신될 때 누적된 시간 값을 초기화할 수 있다.

[0060] SW 와치독(124)이 킥 함수를 수행하는 방법에 대해 도 3을 참조하여 설명하기로 한다. 도 3을 참조하면, Thread B에서 Tread A로 변경되면, 스케줄러(122)는 Thread B 수행 시간을 계산하고, Thread A 시작 시간을 저장한다. 그러면, SW 와치독(124)은 Thread B 수행 시간을 누적하고, 현재 수행 Tread를 A로 저장한다. 그런 후, Thread

A에서 Tread C로 변경되면, 스케줄러(122)는 Thread A 수행 시간을 계산하고, Thread C 시작 시간을 저장한다. 그러면, SW 와치독(124)은 Thread A 수행 시간을 누적하고, 현재 수행 Tread를 C로 저장한다. 그런 후, Thread C에서 Tread A로 변경되면, 스케줄러(122)는 Thread C 수행 시간을 계산하고, Thread A 시작 시간을 저장한다. 그러면, SW 와치독(124)은 Thread C 수행 시간을 누적하고, 현재 수행 Tread를 A로 저장한다. Thread C에서 Tread A로 변경되었을 때, SW 와치독(124)에는 실제 Thread A가 수행된 시간이 누적되어 저장되어 있다. Thread A는 기 설정된 타임아웃 내에 킥 신호를 생성하여 SW 와치독(124)으로 전송할 수 있다. Thread A로부터 킥 신호를 수신한 SW 와치독(124)은 누적 수행 시간을 초기화 함으로써 HW watchdog 처럼 동작할 수 있다. 따라서 침입 감지 목적으로 초기 생성된 각 스레드는 실제 프로세서(130)를 점유한 시간 기준으로 특정 시간(타임아웃) 내에 반드시 킥 신호를 발생시키도록 작성되어야 한다.

[0061] 또한, SW 와치독(124)은 ECU(200)로부터 암호 처리 요청 메시지 수신 시, 밸리데이트 함수를 실행하고, 밸리데이트 함수의 실행 결과 모든 스레드가 정상인 경우, 암호 알고리즘을 동작하며, 암호 처리 요청 메시지에 포함된 시드 값에 대한 처리 결과를 응답 메시지에 포함시켜 ECU(200)로 전송할 수 있다. 밸리데이트 함수의 실행 결과, 적어도 하나의 스레드가 비정상인 경우, SW 와치독(124)은 응답 메시지를 ECU(200)로 전송하지 않을 수 있다.

[0062] ECU(200)는 내부 네트워크 기반으로 암호 처리 요청 메시지를 SW 와치독(124)에 전송하여 밸리데이트(validate) 함수가 수행되도록 하고, 암호 처리 요청 메시지에 대한 응답 메시지를 통해 외부 침입을 판단할 수 있다.

[0063] 즉, ECU(200)는 내부 망 통신 기반으로 암호 처리 요청 메시지를 SW 와치독(124)에 전송하여 밸리데이트 함수가 수행되도록 할 수 있다. 이때, ECU(200)는 암호 처리 요청 메시지에 암호 알고리즘과 시드(seed) 값을 넣어서 SW 와치독(124)으로 전송할 수 있다. SW 와치독(124)은 밸리데이트 함수의 수행 결과를 ECU(200)로 전송할 수 있다.

[0064] ECU(200)는 SW 와치독(124)으로부터 암호 처리 응답 메시지를 수신하고, 그 암호 처리 응답 메시지를 확인함으로써 외부 네트워크 연결 제어기(100)에 외부 침입이 있는지 판단할 수 있다.

[0066] 도 4는 본 발명의 일 실시예에 따른 응용프로그램의 동작을 설명하기 위한 도면이다.

[0067] 도 4를 참조하면, 응용프로그램이 시작되면(S410), 응용프로그램은 초기화하고 병렬 처리를 위한 스레드를 생성한다(S420). 이때, 응용프로그램은 내부 변수 등을 초기화할 수 있다. 응용프로그램이 스레드를 생성하는 방법에 대한 상세한 설명은 도 5를 참조하기로 한다.

[0068] S420 단계가 수행되면, 각 스레드는 반복 구문 안에서 설정된 타임아웃 내에 킥 신호를 생성하며 반복 처리 루틴을 수행한다(S430). 이때, 스레드가 외부 침입에 의하여 제어권을 빼앗겼을 경우 해당 스레드는 킥 신호를 생성하지 않으므로, SW 와치독(124)은 스레드의 이상 유무를 검출할 수 있다. 각 스레드가 타임아웃 내에 킥 신호를 SW 와치독(124)으로 전송하면, SW 와치독(124)은 해당 스레드가 정상이라고 판단할 수 있다.

[0069] S430 단계의 수행으로, 반복 처리 루틴이 완료되면, 각 스레드는 점유 자원을 해제하고(S440), 응용 프로그램이 종료된다(S450).

[0071] 도 5는 본 발명의 일 실시예에 따른 스레드 생성 방법을 설명하기 위한 도면이다.

[0072] 도 5를 참조하면, 스레드 생성이 시작되면(S510), 응용프로그램은 해당 스레드가 SW 와치독(124)이 필요한 스레드인지를 판단한다(S520).

[0073] S520 단계의 판단결과, SW 와치독(124)이 필요한 스레드인 경우, 응용 프로그램은 전체 스레드 수를 '1' 증가시키고(S530), 각 스레드에 설정된 타임아웃 및 비정상 액션을 저장하고, 저장공간을 할당하여 초기화를 수행한다(S540).

[0074] S540 단계가 수행되면, 응용프로그램은 해당 스레드에 스택 메모리를 할당함으로써, 스레드 생성을 완료한다(S550).

[0076] 도 6은 본 발명의 일 실시예에 따른 킥 함수를 설명하기 위한 흐름도이다.

[0077] 도 6을 참조하면, SW 와치독(124)은 현재 수행 중인 스레드로부터 킥 신호의 수신 여부를 판단한다(S610).

[0078] 현재 수행중인 스레드로부터 킥 신호가 수신되면, SW 와치독(124)은 해당 스레드의 누적 수행 시간을 초기화(0)한다(S620).

[0080] 도 7은 본 발명의 일 실시예에 따른 밸리데이트 함수를 설명하기 위한 흐름도이다.

[0081] 도 7을 참조하면, 밸리데이트 함수가 시작되면(S710), SW 와치독(124)은 현재 수행 중인 스레드의 누적 수행 시간이 타임아웃 미만인지를 판단한다(S720).

[0082] S720 단계의 판단결과, 누적 수행 시간이 타임아웃 미만이면, SW 와치독(124)는 스레드의 순번을 '1'증가하고(S730), 1증가된 순번이 총 스레드 수(N) 미만인지를 판단한다(S740).

[0083] S740 단계의 판단결과, 총 스레드 수(N) 미만이면, SW 와치독(124)은 S720 단계를 수행한다.

[0084] 만약, S720 단계의 판단결과, 누적 수행 시간이 타임아웃 미만이 아니면, SW 와치독(124)은 해당 스레드를 비정상 스레드로 판단하고(S750), 해당 스레드에 설정된 비정상 액션을 수행한 후(S760), S730 단계를 수행한다

[0085] 만약, S740 단계의 판단결과, 총 스레드 수 미만이 아니면, SW 와치독(124)은 비정상인 스레드가 존재하는지를 판단한다(S770).

[0086] S770 단계의 판단결과, 비정상인 스레드가 존재하면, SW 와치독(124)은 밸리데이트 함수를 종료한다(S780).

[0087] 만약, S770 단계의 판단결과, 비정상 스레드가 존재하지 않으면, SW 와치독(124)은 모든 스레드가 정상이라고 판단하고(S790), 밸리데이트 함수를 종료한다.

[0088] 본 발명은 응용프로그램 개발할 때 각 스레드가 주기적으로 시스템 함수를 사용하여 킥 신호를 생성하도록 하며, OS 모듈(120)은 SW 와치독(124)을 사용하여 실제 스레드 수행 시간 기반으로 킥 신호 발생을 확인한다면 제어기(100) 내부에서 외부 침입에 의하여 스레드가 비정상 동작하는 것을 검출 및 대응할 수 있다.

[0090] 상술한 바와 같이, 본 발명의 일 측면에 따른 차량 네트워크 침입 탐지 시스템 및 방법은, 제어기(100) 동작 상태에서 SW 와치독을 사용하여 스레드 이상 유무를 확인함으로써, 외부 침입을 감지할 수 있다.

[0091] 본 발명의 일 측면에 따른 차량 네트워크 침입 탐지 시스템 및 방법은, 제어기 내부 SW에 침입이 있는지 SW 와치독을 사용하여 모니터링하므로, 외부 침입 초기부터 감지하여 이상 제어를 방지할 수 있고, 사이버 보안 위협에 보다 강건하다.

[0092] 본 발명의 일 측면에 따른 차량 네트워크 침입 탐지 시스템 및 방법은, 외부 침입자의 SW가 휘발성 메모리(volatile memory)에 상주하는 경우도 감지 가능하므로 외부 침입에 강건한 차량 네트워크를 구성할 수 있다.

[0093] 본 발명의 일 측면에 따른 차량 네트워크 침입 탐지 시스템 및 방법은, 외부 침입에 의하여 이상 동작한 경우뿐만 아니라 SW 버그 등에 의한 이상 동작도 모니터링 가능하므로 최근 점점 더 복잡해지는 차량용 SW 생태계에서 필수적인 고장 감지 및 해결 방법이 될 것이다.

[0094] 본 발명은 도면에 도시된 실시예를 참고로 하여 설명되었으나, 이는 예시적인 것에 불과하며 당해 기술이 속하는 기술분야에서 통상의 지식을 가진 자라면 이로부터 다양한 변형 및 균등한 타 실시예가 가능하다는 점을 이해할 것이다. 따라서, 본 발명의 진정한 기술적 보호범위는 아래의 특허청구범위에 의하여 정해져야할 것이다.

부호의 설명

[0095] 100 : 외부 네트워크 연결 제어기

110 : 프로세스

120 : OS 모듈

122 : 스케줄러

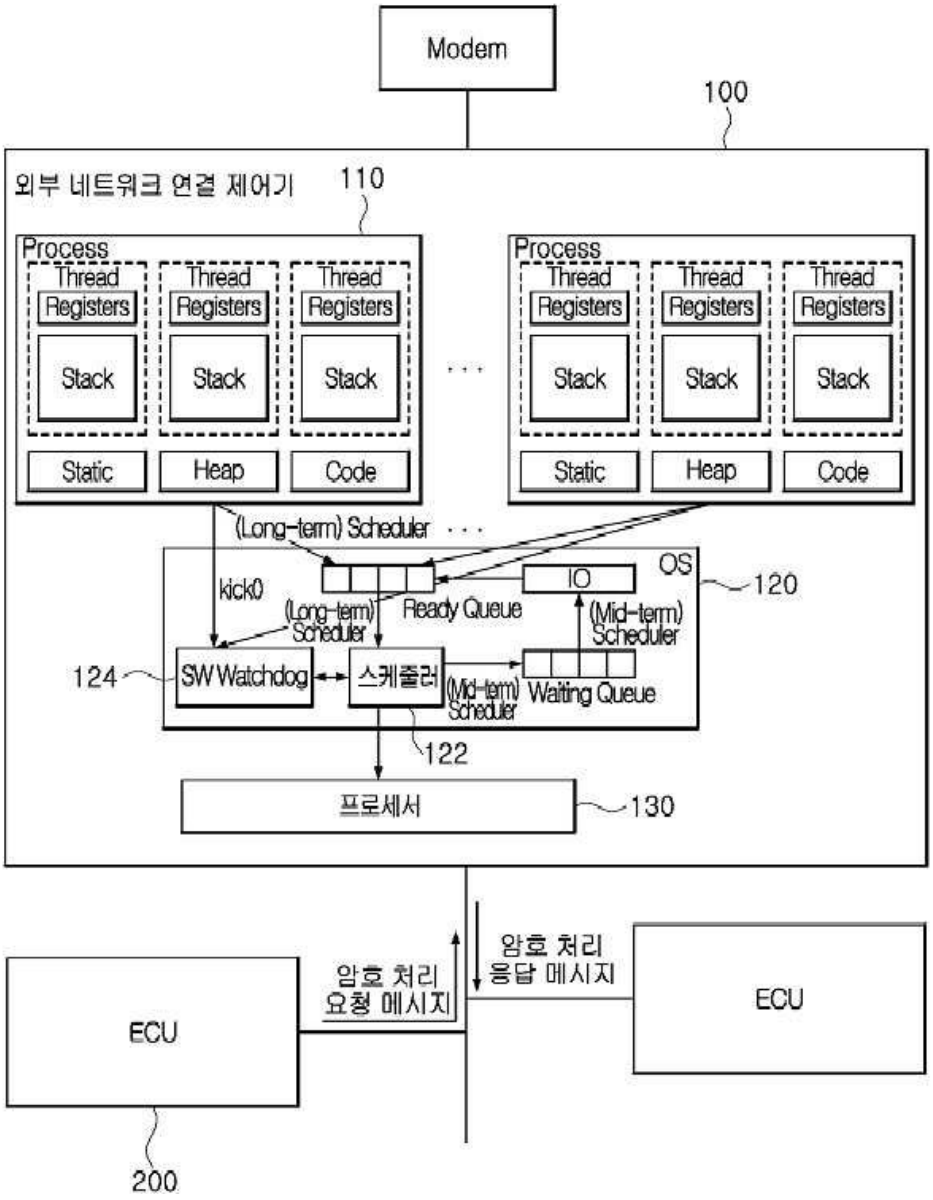
124 : SW 와치독

130 : 프로세서

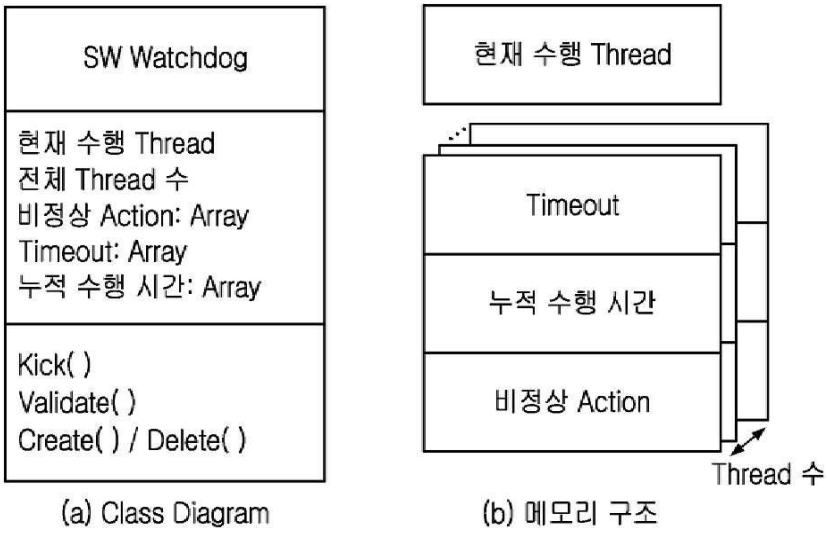
200 : ECU

도면

도면1

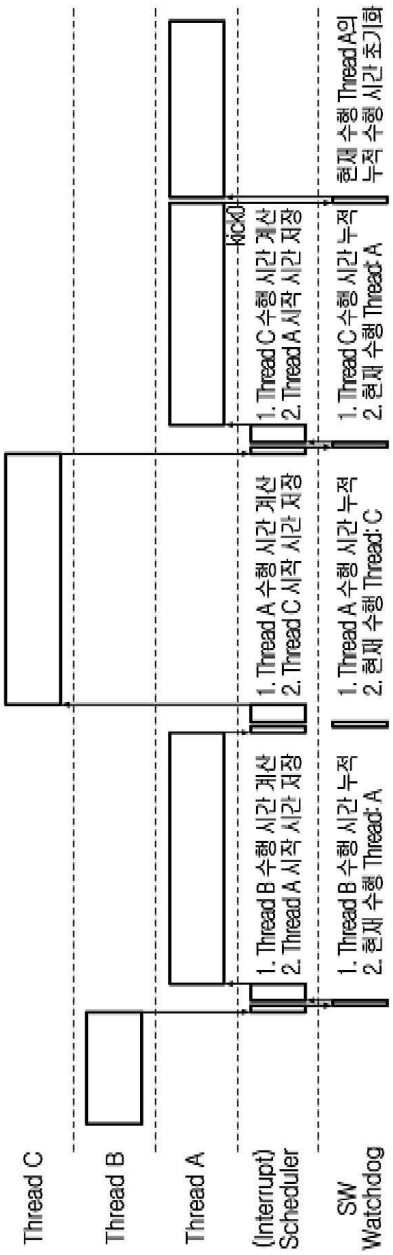


도면2

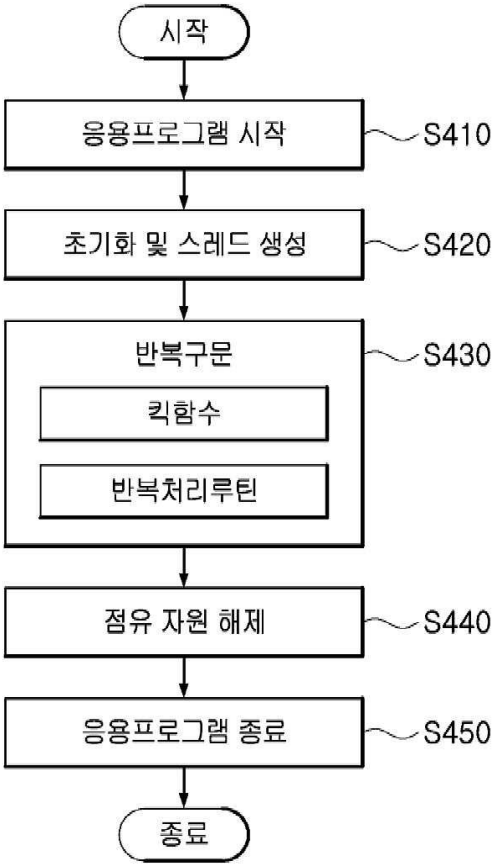


(a) Class Diagram (b) 메모리 구조

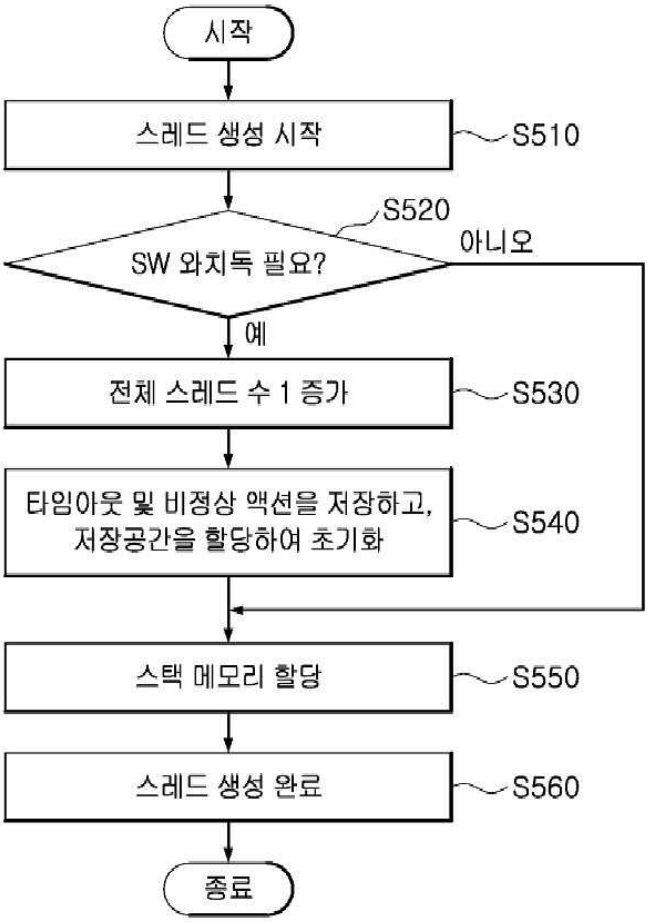
도면3


Thread C
Thread B
Thread A
(Interrupt)
Scheduler
SW
Watchdog
1. Thread B 수행 시간 계산
2. Thread A 시작 시간 저장
1. Thread B 수행 시간 누적
2. 현재 수행 Thread: A
1. Thread A 수행 시간 계산
2. Thread C 시작 시간 저장
1. Thread A 수행 시간 누적
2. 현재 수행 Thread: C
kick()
1. Thread C 수행 시간 계산
2. Thread A 시작 시간 저장
1. Thread C 수행 시간 누적
2. 현재 수행 Thread: A
현재 수행 Thread A의
누적 수행 시간 초기화

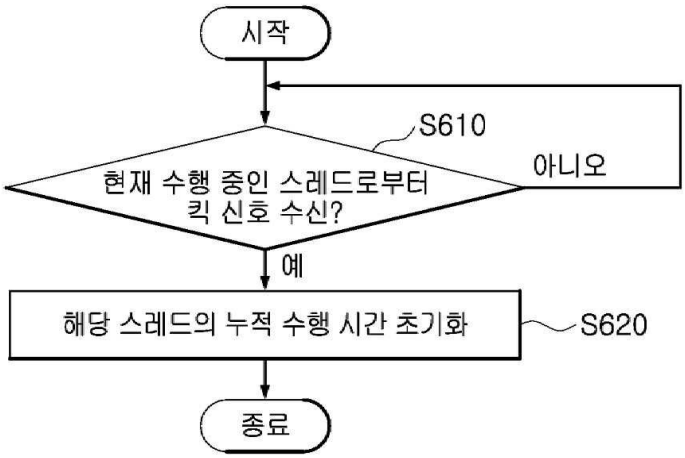
도면4


시작
응용프로그램 시작
S410
초기화 및 스레드 생성
S420
반복구문
S430
킥함수
반복처리루틴
점유 자원 해제
S440
응용프로그램 종료
S450
종료

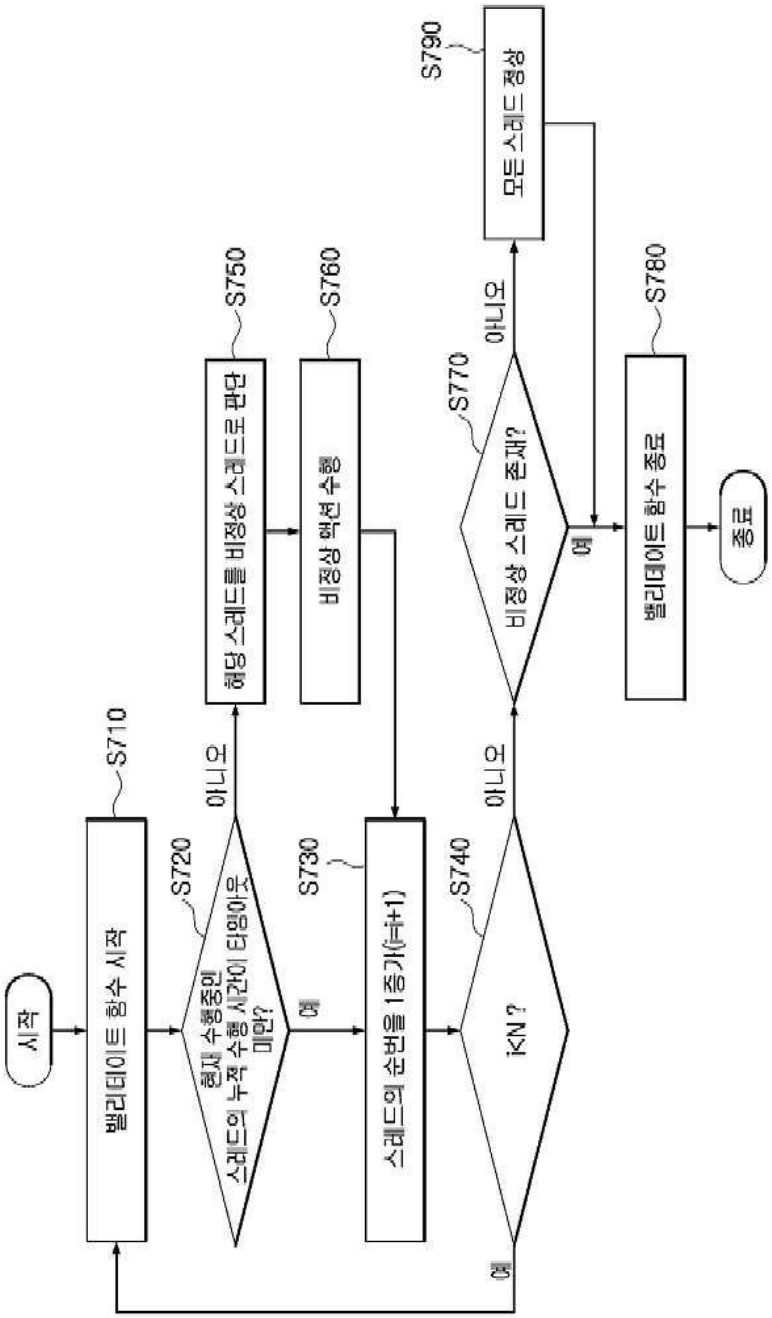
도면5


시작
스레드 생성 시작
S510
S520
SW 와치독 필요?
아니오
예
전체 스레드 수 1 증가
S530
타임아웃 및 비정상 액션을 저장하고,
저장공간을 할당하여 초기화
S540
스택 메모리 할당
S550
스레드 생성 완료
S560
종료

도면6


시작
S610
현재 수행 중인 스레드로부터
킥 신호 수신?
아니오
예
해당 스레드의 누적 수행 시간 초기화
S620
종료

도면7


시작
밸리데이트 함수 시작
S710
현재 수행중인 스레드의 누적 수행 시간이 타임아웃 미만?
S720
아니오
예
해당 스레드를 비정상 스레드로 판단
S750
비정상 액션 수행
S760
스레드의 순번을 1증가(i=i+1)
S730
iKN ?
S740
아니오
예
비정상 스레드 존재?
S770
아니오
예
모든 스레드 정상
S790
밸리데이트 함수 종료
S780
종료