	고등급(EAL6 이상) 보안마이크로커널 개발				
	작성자	이혁	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-10	파일명	2021-기술문서02-SW설계안전성분석보고서-v3.hwp	
	제목	소프트웨어 설계 안전성 분석 보고서			

소프트웨어 설계 안전성분석 보고서

수정일자	수정자	버전	추가/수정 항목	내 용
2020-7-17	박웅섭	1.0		초안작성
2020-11-22	박웅섭	2.0		
2021-12-10	이혁	3.0		

<제목 차례>

<표 차례>

<그림 차례>

I. 목적

본 보고서는 고등급 보안 마이크로 커널(CHAOS)에 필요한 안전기능 설계의 안전성 분석에 관해서 기술하는 것에 그 목적이 있다. 이것을 통해 개발하는 시스템 안전기능의 설계가 안전 목표를 만족하는지에 대해서 분석한다.

II. 범위

본 명세서는 고등급 보안 마이크로 커널(CHAOS)에 필요한 안전기능 중 스케줄 기능과 관련된 모듈을 그 범위로 한다.

III. 용어 정의 및 약어

- 없음

IV. 참고 문헌

- IEC, IEC61508. “61508 functional safety of electrical/electronic/programmable electronic safety-related systems.“ International electrotechnical commission (1998).
- ISO, ISO26262. “26262: Road vehicles-Functional safety.“ International Standard ISO/FDIS 26262 (2011).
- DoD, U. S. “MIL-STD-882C-System Safety Program Requirements.“ US DoD (1993).
- FAA System Safety Handbook. “Federal Aviation Administration.“ (2000).
- 소프트웨어 안전 기능 요구사항 명세서 (2019)
- 소프트웨어 설계 명세서 (2020)
- SW 안전성 공통 개발 가이드. “정보통신산업진흥원.“ (2016).
- ChibiOS/RT The Ultimate Guide (RT 6) https://www.chibios.org/dokuwiki/doku.php?id=chibios:documentation:books:rt:start
- ArduPilot Development Site https://ardupilot.org/dev/index.html

V. 시스템 개요

본 보고서에서 안전성 분석의 대상인 마이크로 커널은 드론을 제어하기 위해 탑재되는 소프트웨어이며 상위수준 시스템 구조는 [그림 1]과 같다.

그림 1 상위수준 시스템 구조

커널 레벨 시스템에는 [그림 2]와 같은 모듈로 구성되어 있다. 세마포어, 뮤텍스, 메시지, 이벤트 등의 모듈은 스케줄러에 의해 동작하며 이진 세마포어와 메일박스는 세마포어에 의해 동작한다.

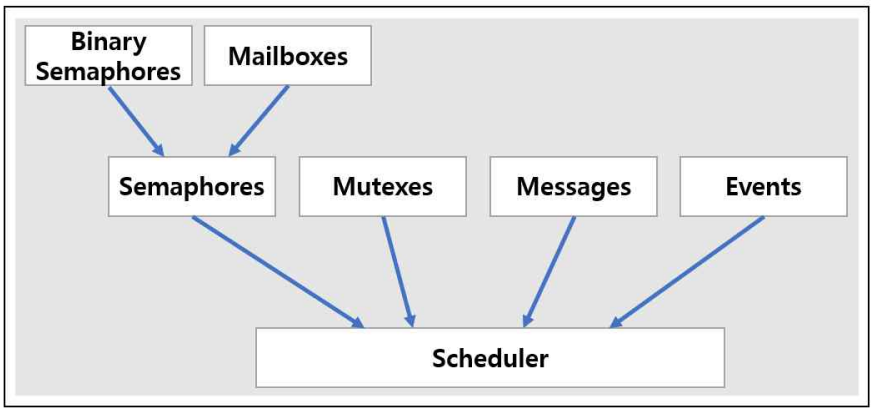

그림 2 CHAOS 커널레벨 시스템 구조

스케줄러 시스템은 [그림 3]과 같이 구성되어 있으며 모든 요소에 대한 시간제한 기능을 구현하기 위해 가상 타이머와 긴밀하게 결합 되어있다.

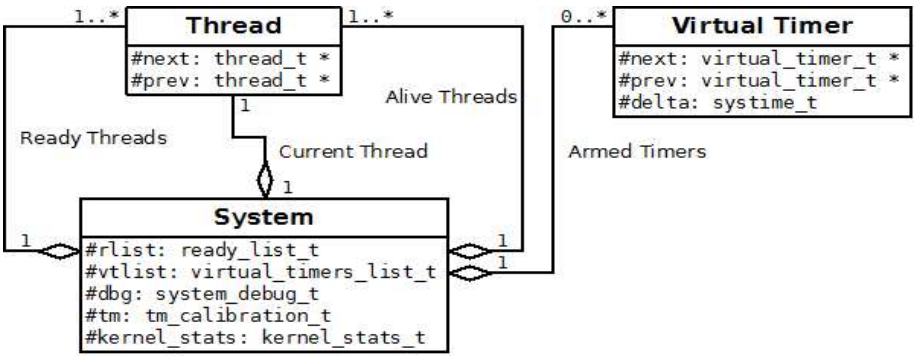

그림 3 CHAOS 스케줄러 시스템 구조

각각의 스레드는 Current, Alive, Ready 상태로 존재하는 스레드로 구분되며 Ready 상태의 스레드는 [그림 4]와 같이 우선순위에 의해 정렬되어 있다. 시스템은 항상 “유휴 스레드”라는 특수 스레드를 실행하며 유휴 스레드를 실행함으로써 시스템 전력 소비를 줄일 수 있으며 유휴 스레드는 오직 Ready, Current 상태에만 있을 수 있고 Sleep, Terminate 상태로 갈 수 없다.

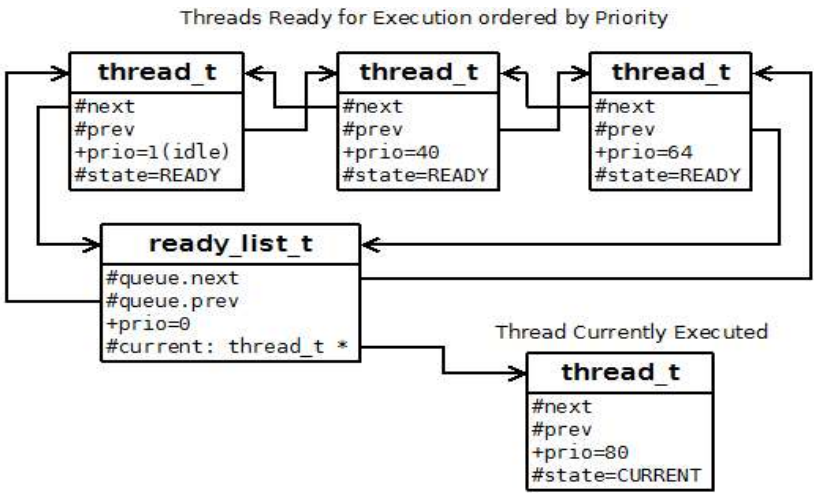

그림 4 CHAOS 스레드 구조

시스템 구조는 아래 [그림 4]와 같다. 드론 하드웨어는 Pixhawk2를 사용한다. 이 장비는 다중 GPS 연결 포트 이외에도 기타 센서 및 장비와 유선 및 무선 통신을 할 수 있도록 Carrier Board를 통해 IC2(Inter Integrated Circuit), CAN(Controller Area Network) 포트와 MAVLink와 같은 무선 통신 프로토콜을 지원하기 위한 Telemetry 포트를 지원한다. 안정성 분석을 수행하는 대상인 Drone OS가 마이크로커널이 탑재되는 영역이다. 실제 비행 제어를 수행하는 시스템은 Flight Computer로 Flight Controller, Drone HW를 담당한다. 그리고 다양한 Application과 Companion Computer OS, SW, HW를 담당하는 영역인 Companion Computer가 있다. 드론 하드웨어는 MAVLink를 사용하여 Companion Computer와 Ground Control Station(GCS)와 통신한다.

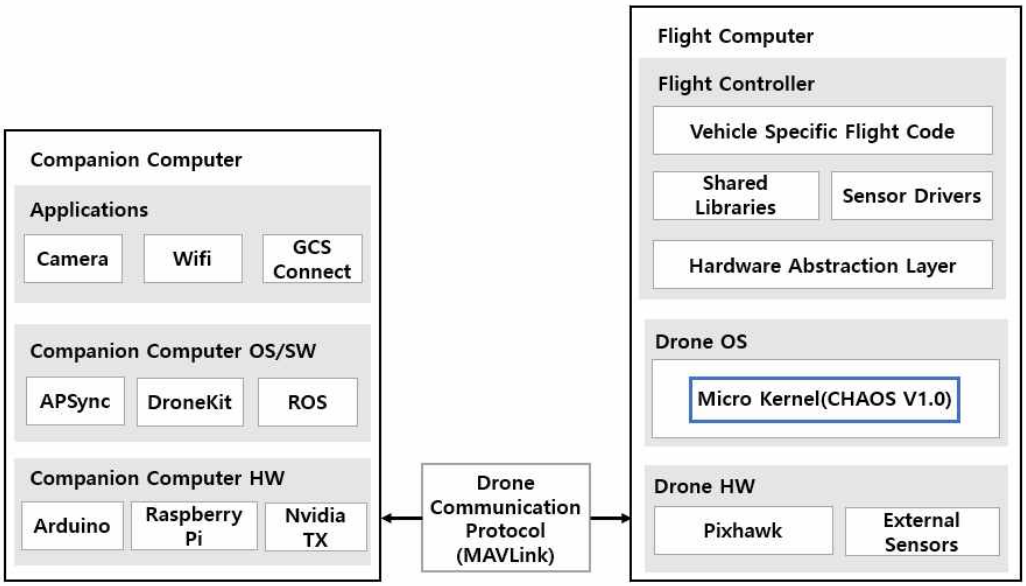

그림 5 소프트웨어 구조설계

VI. 안전성 분석 기법

SW 안전성 분석에 사용된 입력자료는 2019-SW-REQ-안전성 분석서에 제시되어있는 [표 7], [표 8]과 같다.

2019-SW-REQ-안전성 분석서를 기반으로 드론의 안전성을 위협하는 외부적, 내부적 환경 그리고 시스템의 상위 도메인과 하위 도메인에서 발생할 수 있는 위험을 시각적으로 표현한 시나리오는 [그림 6] Fault-tree와 같다. 드론 추락이라는 위험은 연속적인 위험으로 발생한다. 따라서 위험원을 엮어 시나리오를 만들고 시나리오를 체계적으로 구성하기 위해 Fault-tree를 사용한다.

[그림 6] Fault-tree는 2019-SW-REQ-안전성 분석서의 [표 9]에 명시된 것과 같이 각각의 안전 목표는 시나리오에 의해 달성될 수 있으며 S1 ~ S15의 위험원 종류와 매핑되는 것을 볼 수 있다.

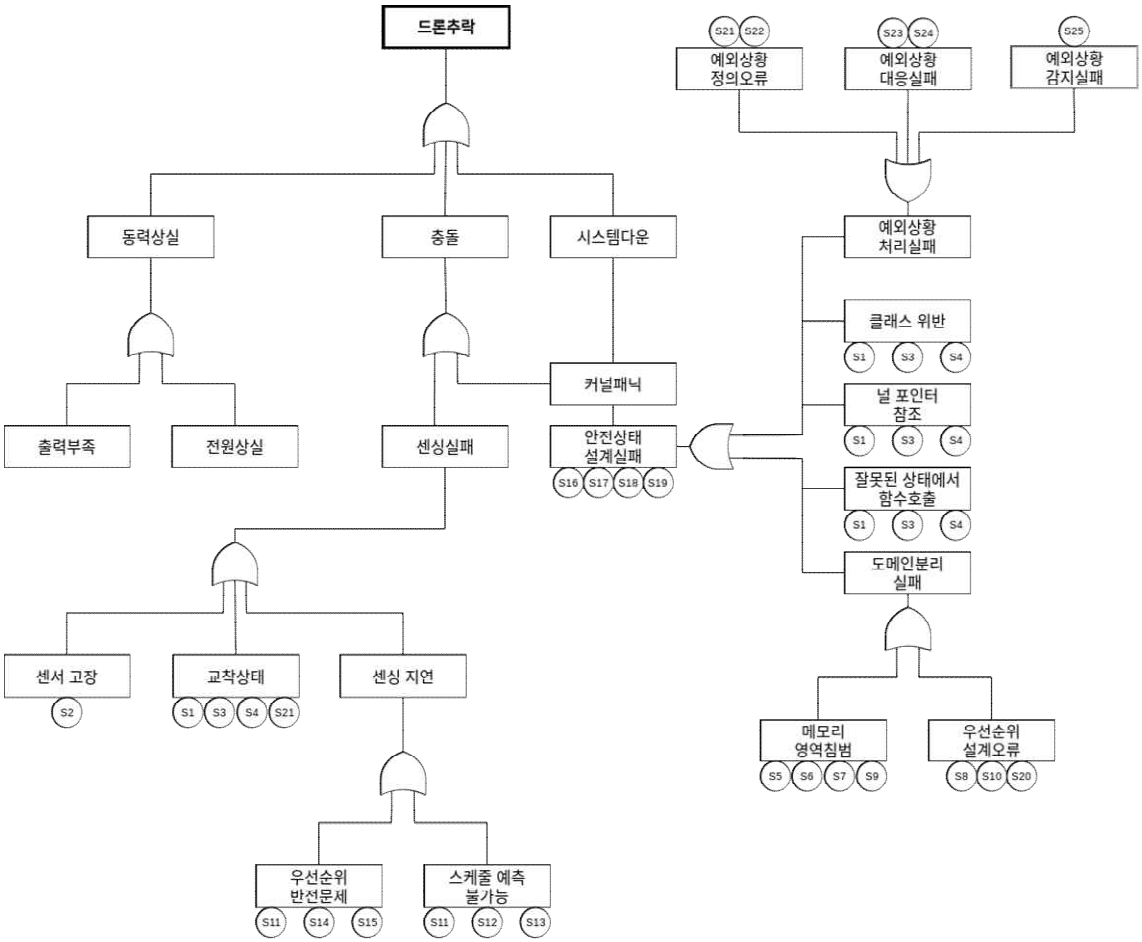

그림 6 드론 추락 관련 Fault-tree

VII. 안전성 분석 결과

2019-SW-REQ-안전성 분석서의 [표 9]에 제시된 안전 목표는 [그림 7], [그림 8], [그림 9]의 Falut-tree 시나리오를 통해 이루어질 수 있으며 각각의 위험원 또한 Fault-tree에 매핑된다. 도출된 시나리오를 통해 2019-SW-REQ-안전성 분석서의 [표 4]를 참조하여 어떠한 시나리오가 더 치명적인가를 판단하여 대책을 마련해야 한다. 안전 목표를 달성하기 위한 대책으로 [표 1]의 안전 기능이 있다. 각 안전 기능은

2019-SW-REQ-안전성 분석서의 [표 4]에 제시된 위험원에 대응된다.

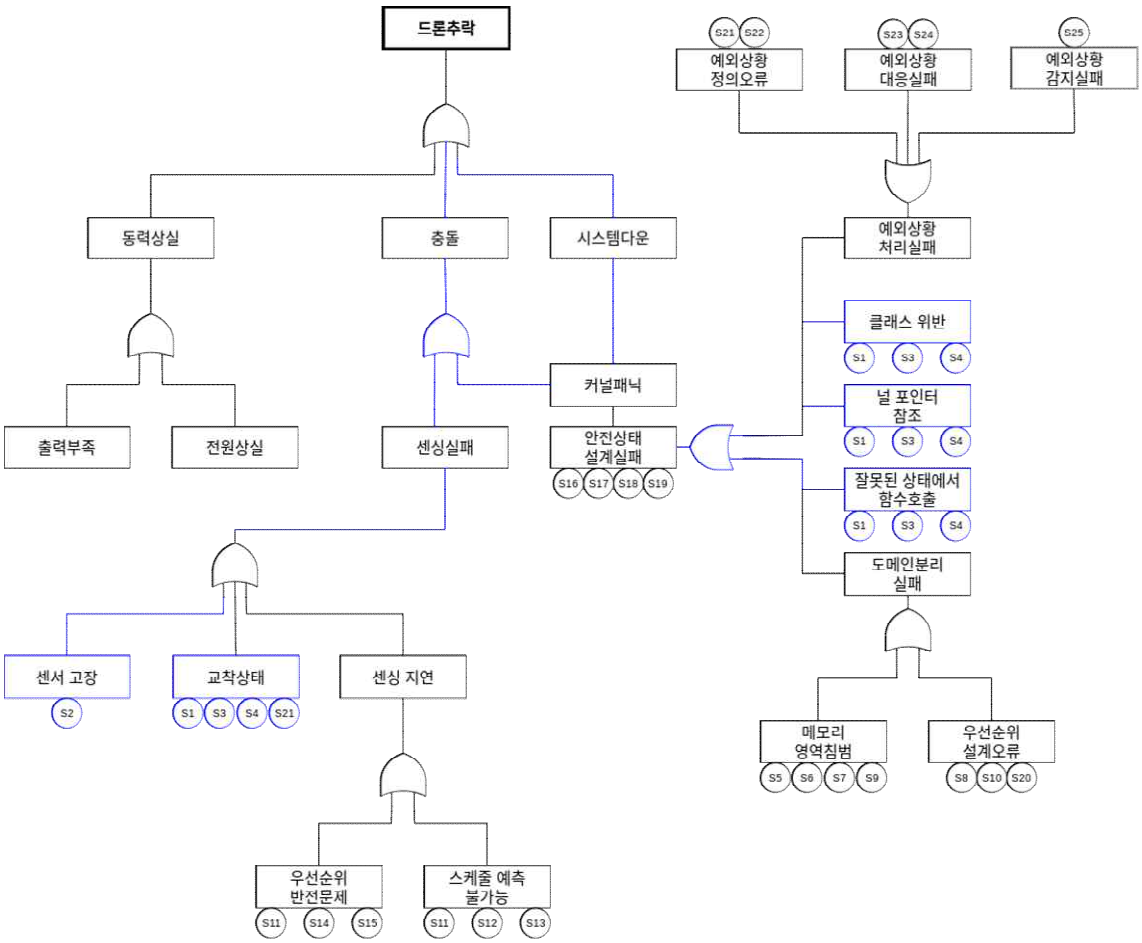

그림 7 SG.1. 자가진단 관련 결함 발생 경로

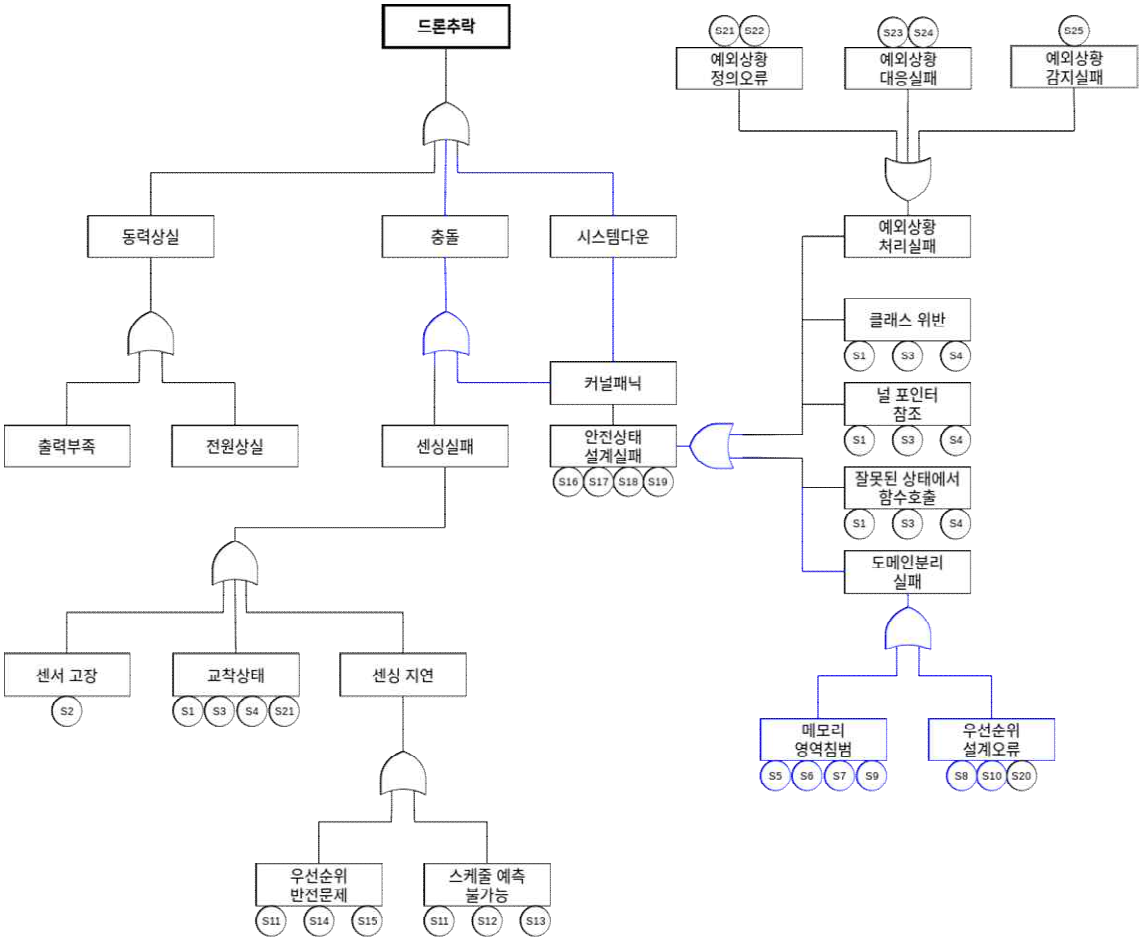

그림 8 SG.2. 도메인 분리 관련 결함 발생 경로

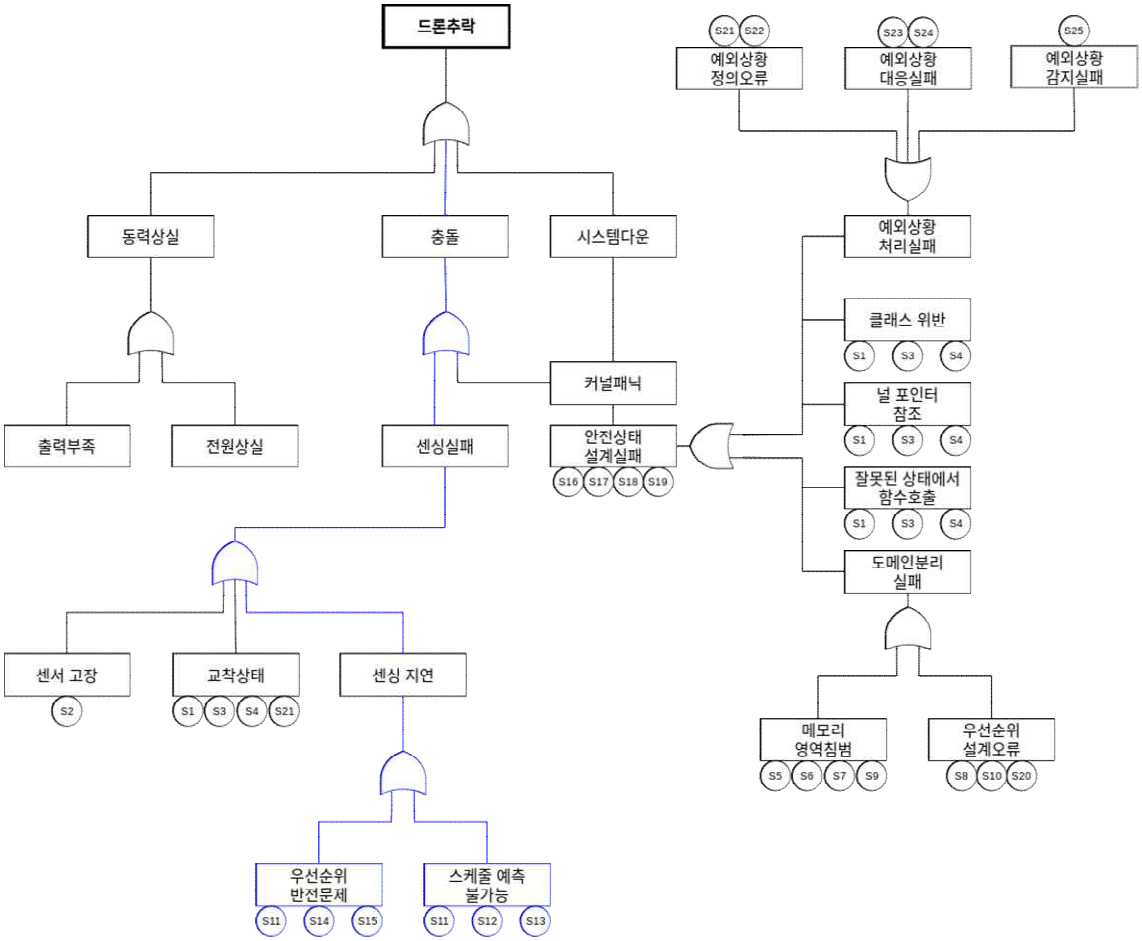

그림 9 SG.3. 스케줄링 예측 가능 관련 결함 발생 경로

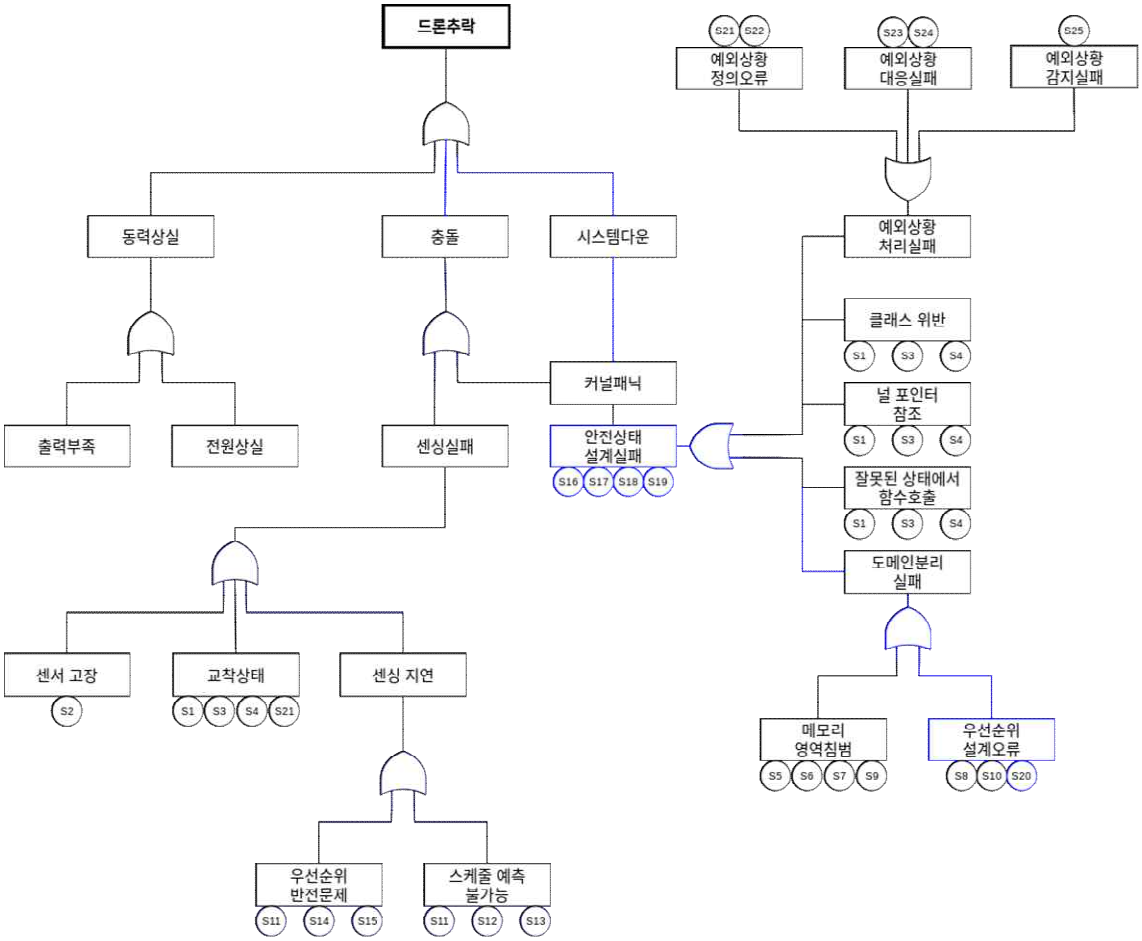

그림 10 SG.4. 안전상태 설계 관련 결함 발생 경로

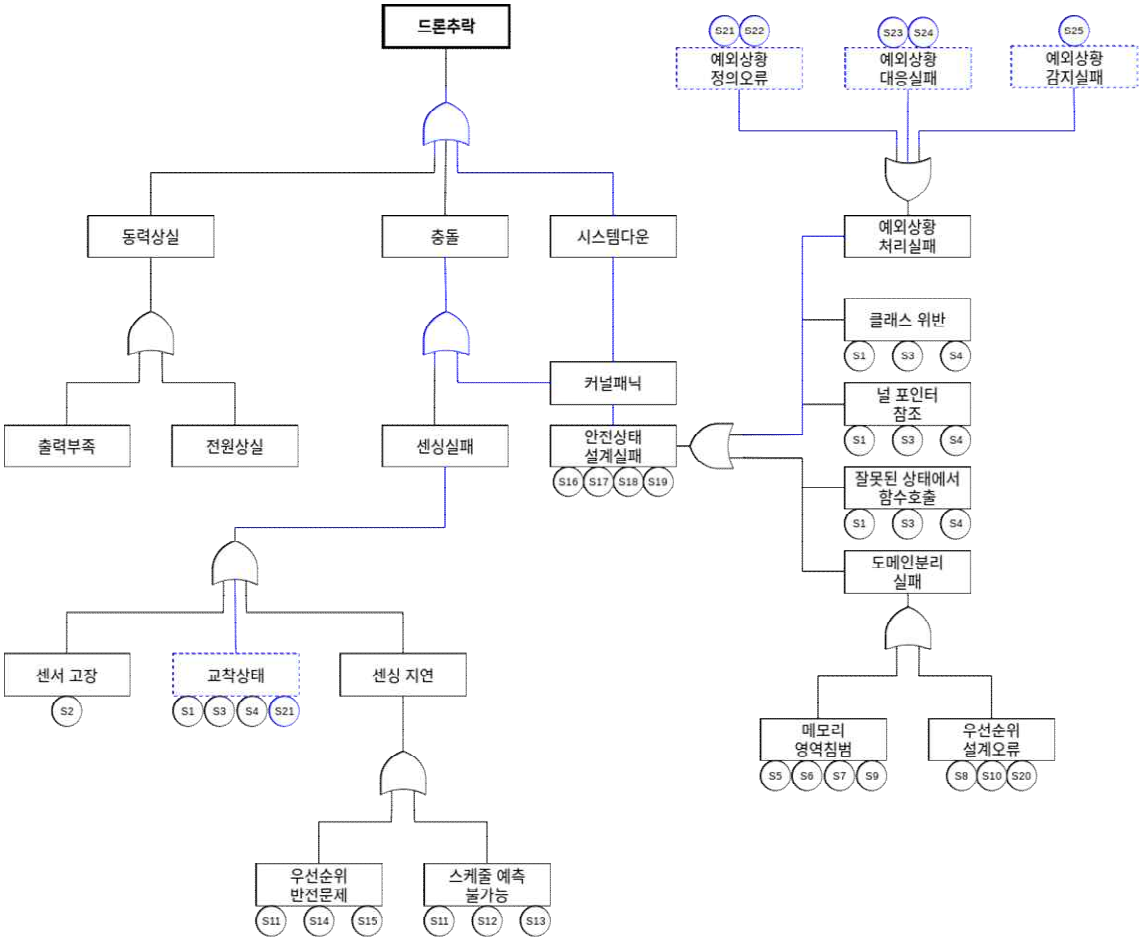

그림 11 SG.5. 예외상황 처리 관련 결함 발생 경로

표 1 위험원 및 대응 안전 기능 매핑

No	위험원 종류	안전 목표	대응 안전 기능
1	커널에 대한 고장이 발생함	SG.1. 자가진단	chDbgCheck() chDbgCheckClassI(), chDbgCheckClassS(), chSysIntegrityCheckI()
2	하드웨어에서 고장이 발생함		
3	모든 고장에 대하여 진단하지 못함		
4	모든 고장에 대하여 적절한 조치를 하지 못함		
5	서로 다른 도메인 사이에서 메모리 조작 및 접근함	SG.2. 도메인 분리	chSysLock(), chSysUnlock()
6	한 개의 도메인에서 발생한 고장이 다른 도메인에게 전파됨		
7	어플리케이션에서 커널 메모리 영역을 조작 및 접근함		
8	어플리케이션의 고장이 커널 모듈에 전파됨		

9	한 개의 어플리케이션이 다른 어플리케이션 메모리 영역을 조작 및 접근함		
10	한 개의 어플리케이션의 고장이 다른 어플리케이션에게 전파됨		
11	특정 프로세스가 CPU 자원을 독점함	SG.3. 스케줄 예측 가능	chThdSleepS(), chSchDoYieldS(), chSchPreemption()
12	특정 프로세스의 작업이 완료될 수 있는지 알 수 없음		
13	특정 프로세스의 작업이 종료되는 시점을 알 수 없음		
14	낮은 우선순위를 갖는 프로세스가 무한 대기 상태에 빠짐		
15	우선순위가 낮은 작업이 우선순위가 높은 작업보다 먼저 실행됨		
16	커널이 제어할 수 없는 상황이 발생함	SG.4. 안전상태 설계	chSysHalt()
17	비정상 상태가 발생했을 때 안전한 상태로 회복하지 못함		
18	특정 요소의 안전상태 회복이 다른 요소의 고장으로 전이됨		
19	특정 요소의 재시작이 다른 요소의 고장으로 전이됨		
20	안전상태로 회복되는 작업보다 다른 작업의 우선순위가 더 높음		
21	정의되지 않은 예외 상황이 발생함	SG.5. 예외 상황 처리	chDbgCheck(), chDbgCheckClassI(), chDbgCheckClassS(), chDbgAssert()
22	부적절한 예외 상황을 정의함		
23	예외 상황에 대하여 대응하지 못함		
24	예외 상황에 대하여 부적절하게 대응함		
25	예외 상황을 감지하지 못함		

표 2 안전 기능 설명

No	안전 기능	기능 설명	대응 위험원
1	chDbgCheck()	커널 고장 진단을 위해 수행되며, 고장 여부를 판별 (진단 내용은 기능 실행 시 조건으로 주어짐)	S1, S2, S3, S4, S21, S22, S23, S24, S25
2	chDbgCheckClassI()	커널 고장이 진단되면, 고장 원인과 기록을 저장하며 시스템 중단 상태로 바꿈	S1, S2, S3, S4, S21, S22, S23, S24, S25
3	chDbgCheckClassS()	시스템이 S-클래스 함수를 호출하기에 적합한 상태인지 확인	S1, S2, S3, S4, S21, S22, S23, S24, S25
4	chSysIntegrityCheckI()	운영체제 자료구조의 무결성을 검사하고, 무결성 검사에 실패 시, 정지기능을 호출	S1, S2, S3, S4
5	chSysLock()	커널의 lock 상태로 이동하여 보호실행을 수행하며, lock 상태에서는 모든 인터럽트를 무력화시키고 보호 코드를 실행	S5,S6,S7,S8,S9,S10
6	chSysUnlock()	커널의 lock 상태에서 빠져나오며, 보호실행과 인터럽트 무력화 기능을 해제	S5,S6,S7,S8,S9,S10
7	chThdSleepS()	호출하는 스래드를 sleep 상태로 보내며, 파라미터로 값만큼의 시스템 틱 시간동안 sleep 상태를 유지	S11,S12,S13,S14,S15
8	chSchDoYieldS()	사용중인 CPU의 시간슬롯을 다음 스래드에게 넘김 (대기 목록에 있는 스래드 중에서 우선순위가 같거나 높은 스래드에게 줄 수 있음)	S11,S12,S13,S14,S15
9	chSchPreemption()	다른 스래드에게 우선권을 넘김	S11,S12,S13,S14,S15
10	chSysHalt()	복구할 수 없는 오류가 감지되었을 때 운영체제에서 호출되며 시스템을 정지	S16,S17,S18,S19,S20
11	chDbgAssert()	검사 조건이 실패하면 커널은 메시지와 함께 정지기능 호출 (조건은 기능 실행 시 주어짐)	S21, S22, S23, S24, S25

