

Software for Safe World

SURESOFT

ALIRA AI

Creating with ALiRA-AI, Validating with VERIFAI The Complete Solution

Test by AI

Test of AI

ALIRA-AITM Powered

ALIRA-AI Hub

SW 공학 AI 모델 컬렉션

ALIRA-AI Agentic

Agentic AI 솔루션

ALIRA-AI Docs

Code-AI-Doc 솔루션

ALIRA-AI Workspace

End-to-End Agentic Automation

VERIFAITM Solutions

VERIFAI-M

AI 모델에 대한 검증 솔루션

VERIFAI-A

Agentic AI 검증 솔루션

VERIFAI-D

AI 시스템 검증을 위한 데이터 솔루션

VERIFAI-V

비전 모델 데이터 증강 솔루션

Creating with ALiRA-AI, Validating with VERIFAI The Complete Solution

Test by AI

ALIRA-AITM Powered

ALIRA-AI Hub
SW 공학 AI 모델 컬렉션

ALIRA-AI Workbench
Agentic AI 솔루션

ALIRA-AI Docs
Code-AI-Doc 솔루션

ALIRA-AI Workspace
End-to-End Agentic Automation

Test of AI

VERIFAITM Solutions

VERIFAI-M
AI 모델에 대한 검증 솔루션

VERIFAI-A
Agentic AI 검증 솔루션

VERIFAI-D
AI 시스템 검증을 위한 데이터 솔루션

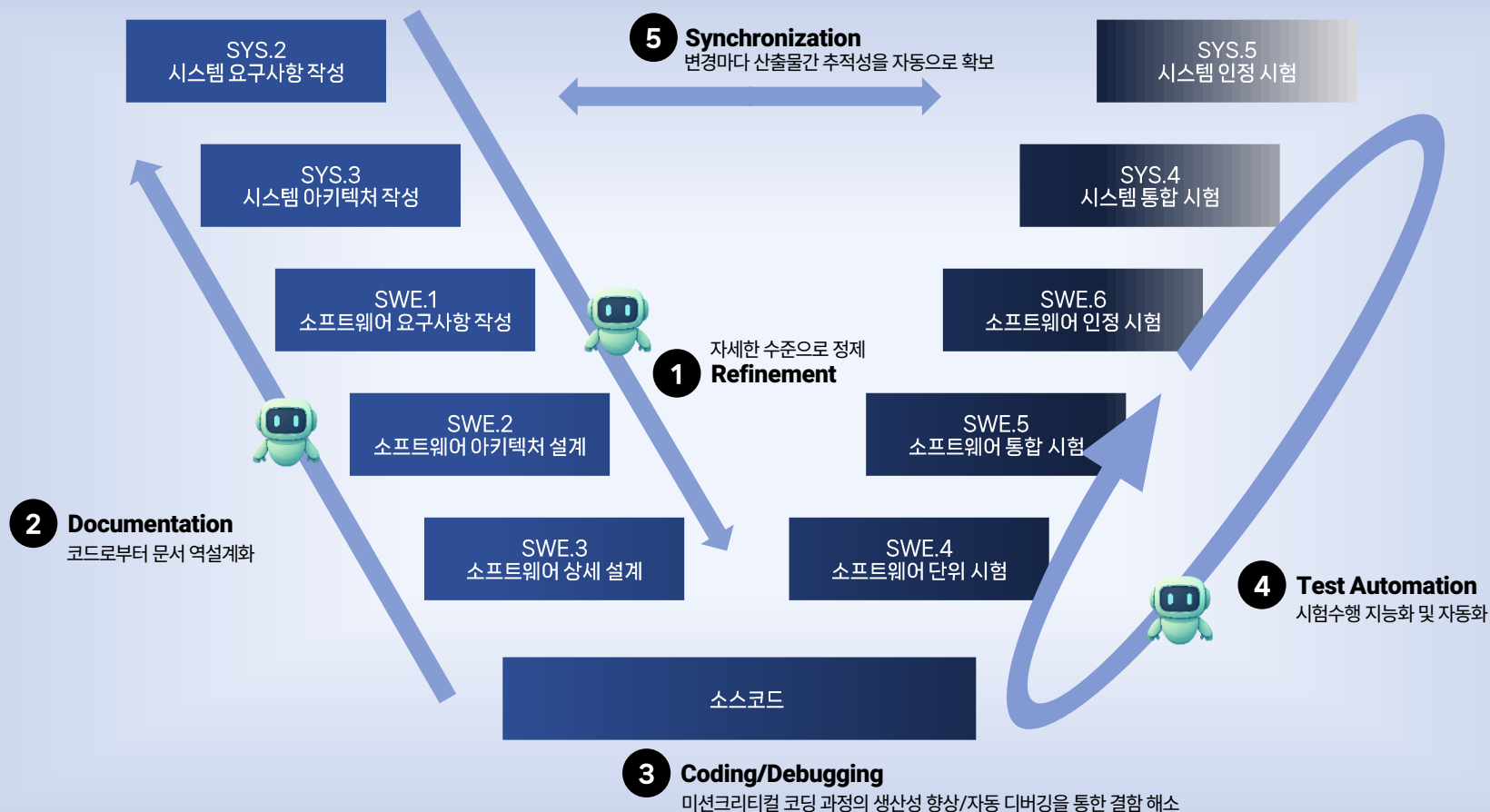
VERIFAI-V
비전 모델 데이터 증강 솔루션

Powered by

ALIRA-AI

Elevating Every Step of Software Engineering

ALIRA-AI는 소프트웨어 공학의 모든 활동에 지능형 자동화 기술을 공급합니다



VISION 2028

ALIRA-AI

workspace

Expert
Knowledge



도메인 지식



ALIRA-AI Agentic

AI 에이전트가 협업하는 계층적 에이전트 시스템

Tools



검증 도구 연계

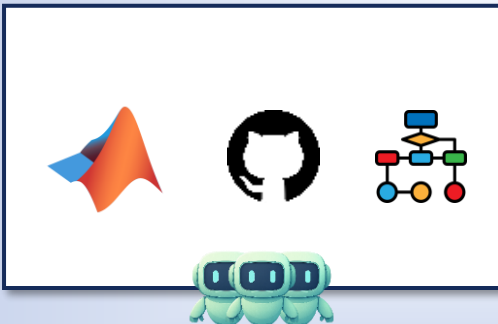
Ontology



요구사항-코드-검증 세속의 연결

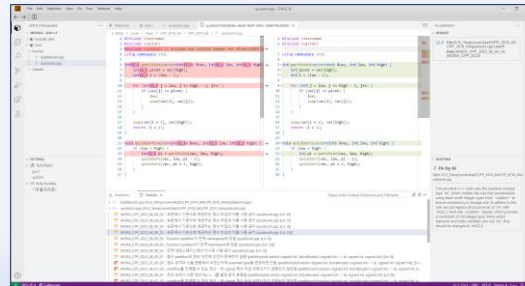
SW 공학 태스크를 위한 AI 에이전트

ALIRA-AI 허브



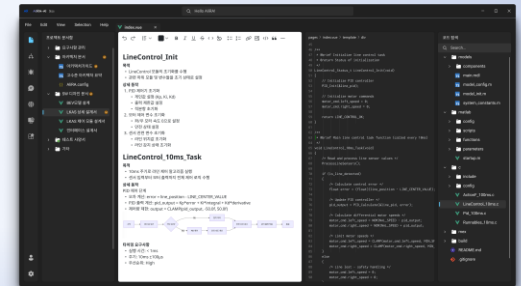
기존 V&V 솔루션의 생산성/지능성 개선

AI-Enabled Test Automation



복합 태스크에 집중된 개발 환경

태스크 특화 생산성 도구



ALIRA-AI workspace

완전 자율 지능형 에이전트 기반 테스트 활동 모니터링, 품질 활동 실행, 결과 분석, 품질 활동 제안

기능

- V&V 목표 달성을 위한 완전 자율 에이전트
- 에이전트 그룹의 작업 수행 과정과 상호작용
- 시스템 레벨 산출물 해석 및 관리 (팀 단위 과업 지원)

The image displays three overlapping screenshots of the ALIRA-AI workspace interface, illustrating its capabilities in project management and task execution.

Top Left Screenshot: Project Overview

- 프로젝트 목록 (Project List):** A section for managing projects, including a button to add a new project.
- 진행중 프로젝트 (In Progress Projects):** A list of active projects with progress bars and status indicators. Examples include ".Jisan-SW 분석용 프로젝트" and ".Jisan-SW 분석용 프로젝트".
- 완료된 프로젝트 (Completed Projects):** A list of completed projects, including ".Jisan-SW 분석용 프로젝트".
- 태스크 포스 (Task Force):** A section for managing task forces, including a list of tasks and their status.

Top Right Screenshot: Task Execution Dashboard

- 작업 상황 보드 (Task Status Board):** A dashboard showing the status of tasks across different stages: Backlog, In progress, Review, and Close.
- 산출물 (Outputs):** A list of outputs generated by the system, including "SDD_v2.docx".

Bottom Screenshot: Task Execution Details

- ① 작업을 선택하세요 (Select a task):** A section for selecting a task to execute, featuring a diagram of the V-model lifecycle (Verification Phase and Validation Phase) and a list of tasks.
- ② 계속 진행하기 위해 다음 정보가 필요합니다 (Next information is needed to continue):** A section for providing additional information to continue the task execution, including a list of tasks and their status.

ALIRA-AI Agent

- 기존 확보
- 2025 하반기 진행 중
- 추후 예정

1 Refinement

- 문서 구조 분석
- 요구사항 양식 검사
- 요구사항 양식 수정
- 요구사항 충돌 탐지
- 요구사항 충돌 수정

2 Documentation

- 함수 명세
- 함수 플로우 차트 생성
- 모델 명세
- 컴포넌트 그룹화
- 컴포넌트 의존성 분석
- 컴포넌트 시퀀스 다이어그램 생성
- 데이터 모델 추출
- 주석 품질 개선
- 사용 시나리오 추출
- 기능 명세

3 Coding/Debugging

- MISRA 위배 자동 수정
- 정적 위배 오탐 검증
- 코딩 Convention 검사
- 코드 포매팅
- 버그 패턴 식별
- 런타임 에러 메시지 분석
- 런타임 에러 수정
- 테스트 에러 분석
- 테스트 에러 수정
- 리팩토링 제안

4 Test Automation

- 요구사항 만족 TC 생성
- 테스트 스크립트 생성
- 테스트 드라이버 에러 수정
- Mock 생성
- 테스트 오라클 생성
- 커버리지 분석
- 단위 테스트 설계서 생성
- TC 중복 제거

5 Synchronization

- 요구사항 변경 영향 분석
- 요구사항 변경 영향 요소 수정
- 설계 변경 영향 분석
- 설계 변경 영향 요소 수정
- 코드 변경 영향 분석
- 코드 변경 영향 요소 수정

6 General

- 지식 기반 Chat
- Agent 작업 스케줄링
- Agent 작업 모니터링
- Agent 오류 탐지
- Agent 오류 복구

ALIRA-AI Hub

New

ALIRA-AI Hub

태스크 특화 UI

Model Req TC Generation test_automation

Generates Test Cases based on the provided requirement inputs.

Requirements Input

File Upload

Requirement_PaddleStep.txt

IO List Input

DOWNLOAD TEMPLATE

File Upload

Additional Context

Custom Prompt

Create test cases

```

1  [패들 변속 가능 여부(PaddleChangeEnable)]
2  1. 다음 조건 중 하나라도 만족할 경우 패들 변속 가능 조건을 만족한다. [REQ-006]
3  1.1 변속이크 펄스가 출력되어야 한다. [REQ-006-01]
4  1.2 차량이 30km/h 이상이어야 한다. [REQ-006-02]
5  1.3 크루즈가 활성화되어 있고 SCC가 설정되어 있어야 한다. [REQ-006-03]
6  1.4 변속 모드가 D단이어야 한다. [REQ-006-04]
7  [패들 Step Change 여부(PaddleStepChange)]
8  1. 패들 변속이 가능한 경우 패들 Step Change가 가능 하다. [REQ-007]
9

```

TC ID	TC DESCRIPTION	PRECONDITION	TEST INPUT	EXPECTED VALUE
1	보이지크 패달만 활성화된 경우 패들 변...	-	BrakePedal: true, Speed: 20km/h, ...	PaddleChangeEnable: true
2	차속 30km/h 이상인 경우 패들 변속 ...	-	BrakePedal: false, Speed: 35km/h, ...	PaddleChangeEnable: true
3	크루즈 활성화 + SCC 설정된 경우 패들 ...	-	BrakePedal: false, Speed: 20km/h, ...	PaddleChangeEnable: true
4	D단만 활성화된 경우 패들 변속 가능 여...	-	BrakePedal: false, Speed: 20km/h, ...	PaddleChangeEnable: true
5	모든 조건 미만족 시 패들 변속 불가함 ...	-	BrakePedal: false, Speed: 20km/h, ...	PaddleChangeEnable: false
6	패들 변속 가능 시 패들 Step Change 가...	-	PaddleChangeEnable: true	PaddleStepChange: true

[App Request](#)

[Result Download](#)

기능

- V&V 주요 태스크별 활용 가능한 AI Service 탐색
- 각 AI Service에 대한 독립 UI 제공
- AI Service 활용 가이드 및 유즈케이스 제공

Why

- 고객의 ALIRA 활용성에 대한 Feasibility 확인
- 고객의 ALIRA 단순 사용 시나리오 지원

AI 모델(sLM)을 활용한 Simulink 모델 기반 기능 사양서 자동 생성 솔루션 소개

- 국제 표준 대응과 모델 산출물에 대한 효율적이고 안정적인 사양서 확보를 위한 사양 자동 생성 솔루션 활용 방안 제시
- AI를 활용하여 Simulink 모델 분석을 통해 SDD 수준의 요구사항 자동 추출 및 사양서 자동 생성 가능

필요성

4.4.3. SWE.3 Software Detailed Design and Unit Construction

Base practices

SWE.3.BP1: Develop a detailed design. Develop a detailed design for each software unit defined in the software architectural design that specifies software units with respect to functional and non-

8.5 Work products

8.5.1 Software unit design specification resulting from requirements 8.4.2 to 8.4.5.

NOTE: In the case of model-based development, the software unit design specification is the software unit design and supporting descriptive documentation, using methods listed in 8.4.2 to 8.4.5.

8.4.2 The software unit design and supporting descriptive documentation shall be:

- be suitable to satisfy the software unit design and supporting descriptive documentation required ASIL;
- be consistent with the software architectural design specification; and
- be consistent with the hardware-software interface specification, if applicable.

- 기능 안전 분석(FMEA/FTA)을 위해 StateChart를 이용한 시각화 자료 필요
- A-SPICE, ISO26262 대응 시 요구사항-설계-코드 간 추적성 확보 필요
- 안전 관련 SW는 문서화 및 리뷰가 필수이나 수작업으로 수행 시 휴먼 에러 발생

기대 효과

효율적이고 안정적인 사양 작성 필요

- ☑ 변경 요구사항에 대한 신속하고 정확한 산출물 반영
- ☑ 수작업(누락, 중복, 오기)으로 인한 휴먼 에러 방지
- ☑ 많고 복잡한 모델 대상의 사양 작성/검토에 대한 공수 효율화

솔루션 소개



ALIRA-AI를 활용한 '코드 기반 요구사항 명세 자동 생성 기능' 소개

- A-SPICE, ISO26262 등 인증 기준 대응을 위한 산출물 생성 자동화로 시간/비용 절감 및 문서 품질의 일관성 확보
- AI를 활용하여 소스 코드 분석을 통해 SDD 수준의 요구사항 및 Control Flow Graph 자동 추출 및 문서 자동 생성 가능

필요성

4.4.3. SWE.3 Software Detailed Design and Unit Construction

Base practices

SWE.3.BP1: Develop a software detailed design. Develop a detailed design for each software unit defined in the software architectural design that specifies software units with respect to functional and non-

8.5 Work products

8.5.1 Software unit design specification resulting from requirements 8.4.2 to 8.4.5.

NOTE In the case of model-based development, the software unit design specification is a model and supporting descriptive documentation, using methods listed in 8.4.2 to 8.4.5.

- 8.4.2** The software unit design and the software unit design specification shall be:
- a) be suitable to satisfy the software unit design specification, and
 - b) be consistent with the software architectural design specification, and
 - c) be consistent with the hardware-software interface specification, if applicable.

- A-SPICE, ISO26262 대응 시 산출물 및 추적성 확보에 과도한 리소스 소모
- 코드 중심의 개발 환경으로 문서화가 사후적으로 진행, 문서 간 정합성 확보의 한계 발생
- 반복적 문서 작성 작업으로 생산성 저하, 기존 코드 및 산출물의 효과적 재활용 미흡

기대 효과



<리소스 절감>



<문서 정합성 확보>



<휴먼 에러 방지>

- 반복적인 사양서 작성 업무를 AI를 통해 자동화 함으로써 소요 시간 및 인력 부담 감소
- 코드 기반으로 만들어졌기 때문에 명확한 요구사항 및 문서 정합성 확보 가능
- AI를 통해 사양서를 생성함으로써 휴먼 에러 발생 가능성 감소

솔루션 소개

```
#include <stdio.h>

void controlElectricCar(int batteryLevel, int speed, int temperature) {
    printf("전기차 상태 모니터링 및 제어\n");

    // 배터리 레벨 확인
    if (batteryLevel < 20) {
        printf("경고: 배터리 부족! 충전이 필요합니다.\n");
    }
    else {
        printf("배터리 상태 양호\n");
    }

    // 속도 제어
    if (speed > 120) {
        printf("경고: 과속! 속도를 줄이세요.\n");
    }
    else if (speed < 0) {
        printf("오류: 잘못된 속도값\n");
    }
    else {
        printf("현재 속도: %d km/h\n", speed);
    }

    // 온도 관리
    if (temperature > 80) {
        printf("경고: 과열! 냉각 시스템 가동\n");
    }
    else if (temperature < 0) {
        printf("경고: 저온! 배터리 예열 필요\n");
    }
    else {
        printf("온도 정상\n");
    }
}
```

[소스 코드]

코드 분석 → 자연어 기반
요구사항 추출

ALIRA-AI

코드 분석 → Mermaid
형식의 스크립트 추출

1. 전기차 상태 모니터링 및 제어를 출력해야 합니다.
2. 배터리 레벨을 확인해야 합니다.
 - 2.1 batteryLevel이 20 미만일 경우 다음을 수행해야 합니다.
 - 2.1.1 "경고: 배터리 부족! 충전이 필요합니다."를 출력해야 합니다.
 - 2.2 batteryLevel이 20 이상일 경우 "배터리 상태 양호"를 출력해야 합니다.
3. 속도 관리
 - 3.1 speed가 120보다 클 경우 다음을 수행해야 합니다.
 - 3.1.1 "경고: 과속! 속도를 줄이세요."를 출력해야 합니다.
 - 3.2 speed가 0 미만일 경우 "오류: 잘못된 속도값"을 출력해야 합니다.
 - 3.2.1 "오류: 잘못된 속도값"을 출력해야 합니다.
4. 온도 관리
 - 4.1 temperature가 80보다 클 경우 다음을 수행해야 합니다.
 - 4.1.1 "경고: 과열! 냉각 시스템 가동"을 출력해야 합니다.
 - 4.2 temperature가 0 미만일 경우 "오류: 잘못된 온도값"을 출력해야 합니다.
 - 4.2.1 "오류: 잘못된 온도값"을 출력해야 합니다.

[요구사항 추출]

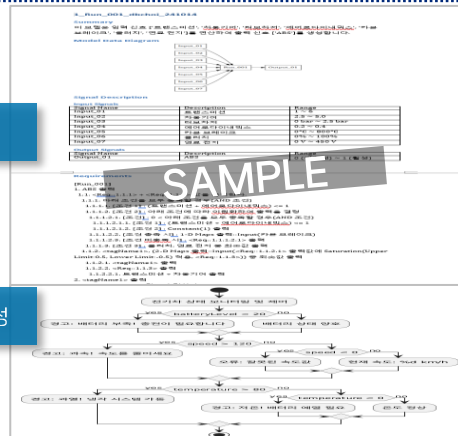
```
graph TD
    A[전기차 상태 모니터링 및 제어] --> B[batteryLevel < 20]
    B -- yes --> C[경고: 배터리 부족! 충전이 필요합니다.]
    B -- no --> D[배터리 상태 양호]
    C --> E[ ]
    D --> E
    E --> F[speed > 120]
    F -- yes --> G[경고: 과속! 속도를 줄이세요.]
    F -- no --> H[ ]
    G --> I[ ]
    H --> I
    I --> J[speed < 0]
    J -- yes --> K[오류: 잘못된 속도값]
    J -- no --> L[현재 속도: %d km/h]
    K --> M[ ]
    L --> M
    M --> N[ ]
    N --> O[temperature > 80]
    O -- yes --> P[경고: 과열! 냉각 시스템 가동]
    O -- no --> Q[ ]
    P --> R[ ]
    Q --> R
    R --> S[temperature < 0]
    S -- yes --> T[오류: 잘못된 온도값]
    S -- no --> U[온도 정상]
    T --> V[ ]
    U --> V
    V --> End[ ]
```

[Mermaid 스크립트]

요구사항 → 문서화

ALIRA-AI

스크립트 → Chart 생성



코드 기반 요구사항 문서 생성 완료

ALIRA-AI를 활용한 '요구사항 기반 Testcase 자동 생성 기능' 소개

- 요구사항 문서를 분석하고, 요구 조건을 충족하는 구조화된 테스트케이스를 자동으로 생성
- AI를 활용하여 입력된 요구사항 정보로부터 Testcase 자동 생성 가능

필요성



사양 변경



테스트케이스 개발



유효성 확인



검증에 적용

- 기능 사양 변경 시 테스트 항목의 유효성 확인 및 신속한 검증 대응이 필요함
- 수동 작성된 테스트케이스 간의 양식 불일치 및 누락 문제 발생 가능
- 요구사항 문서의 해석 오류나 누락을 방지하고, 테스트 항목의 추적성과 일관성 확보 필요

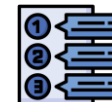
기대 효과



사양 변경 시 TC 즉시 확보



표준화된 양식 적용



요구사항 번호 부여

- 수정된 모델에 대한 테스트케이스 즉시 확보하여 신속한 검증 대응 가능
- 표준화된 형식으로 테스트케이스 일관성 유지하여 작업자간 이해도 향상
- 자동 생성 과정에서 요구사항 번호를 부여하여 ALM 시스템과 연동 가능

솔루션 소개



[양식에 맞게 작성된 요구사항 정보 입력]

ALIRA-AI를 활용한 'Simulink 모델 기반 Testcase 자동 생성 기능' 소개

- 신규/개선 로직의 신속한 검증 대응을 위한 테스트케이스 개발 효율화 방안
- AI를 활용하여 Simulink 모델 분석을 통해 일정한 포맷의 Testcase 자동 생성

필요성



신규/개선 기능 발생



테스트케이스 개발



유효성 확인



검증에 적용

- 제어로직 변경 시 변경점에 대한 신속한 유효성 판단 필요
- 테스트케이스 작성 양식으로부터의 혼선 방지를 위해 표준 양식 필요
- 수정된 로직에 대한 체계적인 테스트케이스 관리 필요

기대 효과



모델 변경 시 TC 즉시 확보



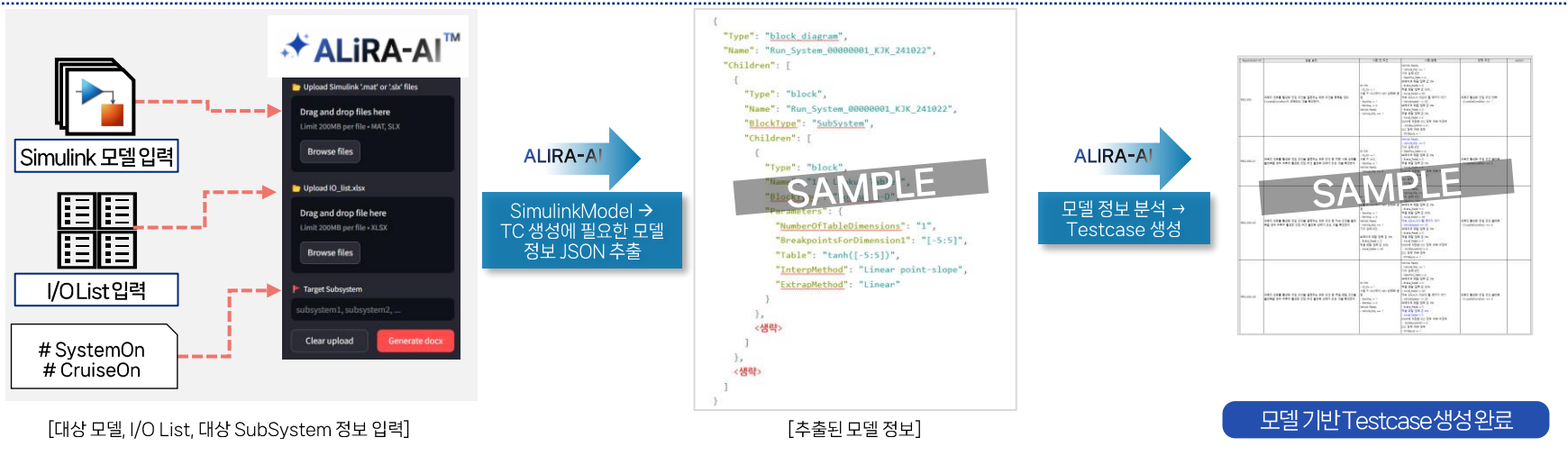
표준화된 양식 적용



요구사항 번호 부여

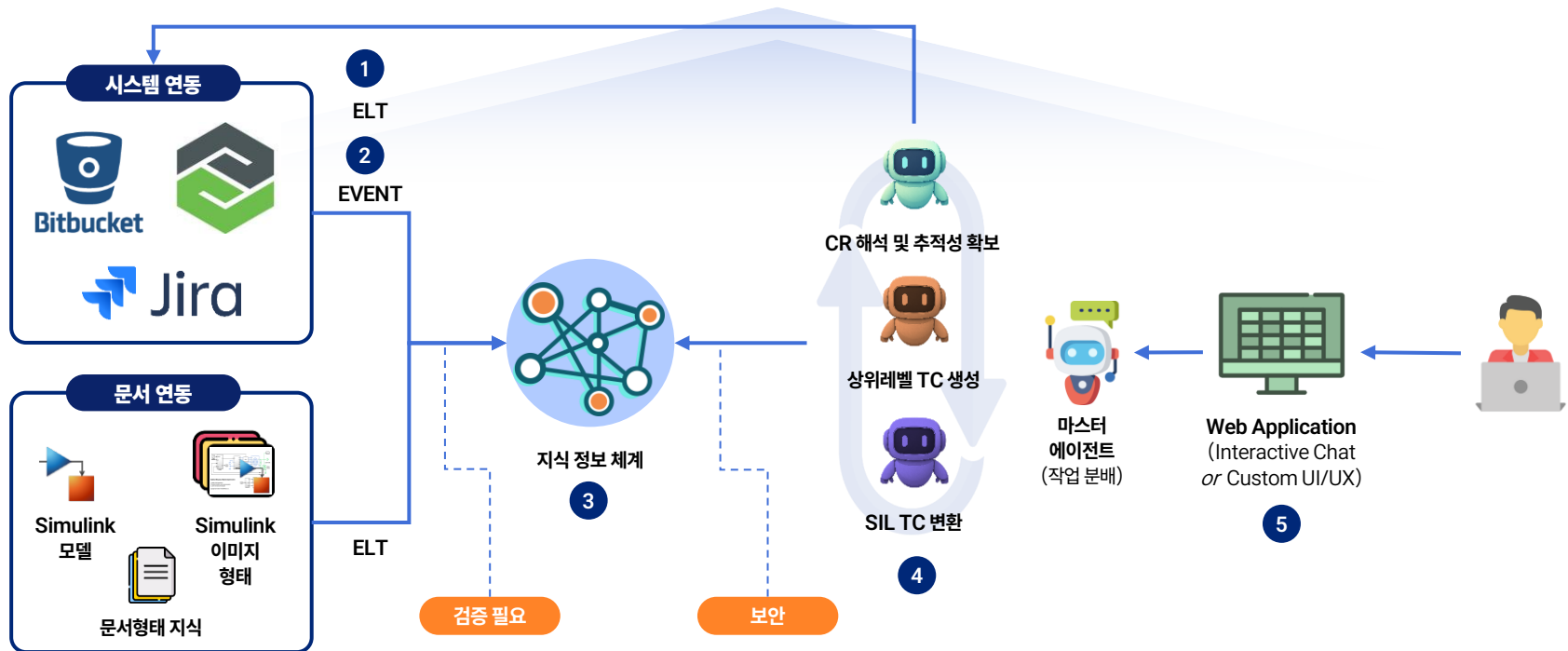
- 수정된 모델에 대한 테스트케이스 즉시 확보하여 신속한 검증 대응 가능
- 표준화된 형식으로 테스트케이스 일관성 유지하여 작업자간 이해도 향상
- 자동 생성 과정에서 요구사항 번호를 부여하여 ALM 시스템과 연동 가능

솔루션 소개



ALIRA-AI Workbench

부품사 CR(change request) 발생시 → 관련 TC 추출, 필요시 신규 TC 생성, SIL TC 변환 → 테스트 수행 → 결과 리포트 정리



1 ELT 처리 (Extract-Load-Transform)
정보시스템에서 As-Is 데이터를 적재 AI 처리를 통해 가공하여 지식 정보 체계에 저장하는 비즈니스 로직

핵심 요소

- 복잡한 문서 형태의 지식(다양한 포맷) 지원
- 문서에 대한 OCR 필요
- Vision 모델을 통한 이미지 해석 필요

2 EVENT
정보시스템의 상태변경 이벤트를 자동으로 감지하여 정해진 지식정보 체계를 자동으로 업데이트하는 처리기 개발

3 지식 정보 체계
AI가 태스크를 수행할 수 있도록 정보간의 관계를 설정하고 제약 조건들이 명시된 정보 체계 개발

4 ALIRA AI
자동화된 지능형 태스크 수행을 위한 에이전틱 AI 솔루션인 ALIRA-AI에 대한 커스터마이징 개발 수행

CR 해석 및 추적성 확보
CR과 관련한 요구사항을 조사하고, 새로운 ID 부여, 추적성 평가 수행

상위레벨 TC 생성
CR과 관련한 TC를 조사하고, 필요한 TC 명세서를 새로 생성

SIL TC 변환
TC 명세서를 기반으로 특정 스크립트로 변환 수행(지정 포맷 검증)

5 Application
사용자 편의성 확보를 위해, 시스템을 운영하고 관리하기 위한 대시보드 성격의 응용 프로그램 개발

The Loop: Read, Reason, Act

무결한 데이터 순환의 힘

1. Ingestion Layer (Write)



Processor

2. Knowledge Core (SSOT)



Ontology

3. Application Layer (Read & Act)

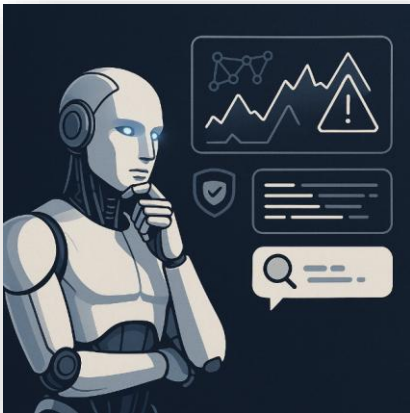


OSDK

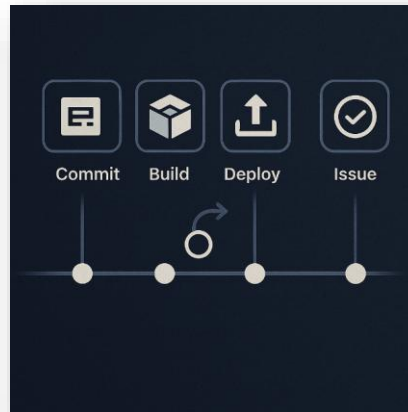
ALIRA-AI

Workbench

1. AI 에이전트 : 분석/추천



2. 시간축 추적 : 인과관계 & 히스토리



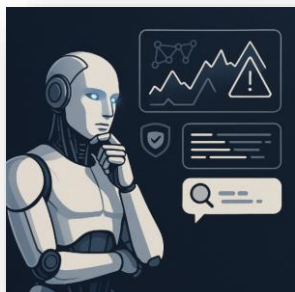
3. 자동화 액션 : 자동화



ALIRA-AI

Workbench

1. AI 에이전트 : 분석/추천



- 그래프 패턴/데이터 기반 분석으로 버그 발생 가능성·품질 리스크 예측
- 코드 리뷰/리팩토링/보안 취약점 자동 탐지 및 추천
- 자연어 질의 응답 기반 액션 실행
- 패턴 발견: 버그 패턴, 성능 이슈 패턴, 개발자 스타일 분석
- 예측 모델: 버그 발생 확률, 코드 품질 예측
- 지능형 추천: 적절한 리뷰어, 유사 이슈 탐색
- 영향도 분석: 코드 변경이 미치는 영향 범위 계산

000 관련 테스트들은 최근에 다 통과했어?

PALADIN

지난 3회 빌드 결과 중 2회는 모두 통과했으나, integration/PaymentFlowIT.testPaymentFailureRollback() 은 2024-07-10 빌드에서 실패했습니다. 실패 원인은 DB 트랜잭션 롤백 타임아웃으로 분석됩니다.

그럼 내가 지금 확인해야 할 가장 중요한 액션은 뭐야?

PALADIN

DB 트랜잭션 타임아웃 설정을 30초 → 60초로 조정하는 것을 추천합니다. 해당 테스트 실패 내역을 JIRA-256 이슈에 자동 코멘트로 기록할까요?

요구사항 "사용자 로그인 시 2FA 인증 추가"는 언제부터 코드에 반영된 거야?

PALADIN

commit a1b23c (24.06.10)에 해당 기능이 처음 구현되었으며, JIRA-142 이슈와 연결되어 있습니다. 이후 commit d4e56f (2024.06.15)에서 리팩터링이 진행되었습니다.

최근 이 기능에서 발생한 이슈나 리스크는 뭐가 있어?

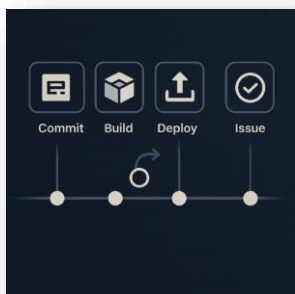
PALADIN

지난 한 달간 로그인 모듈 관련 버그 리포트 2건이 접수되었습니다. 그중 하나는 잘못된 OTP 입력 시 예외 처리가 누락된 문제였으며, 품질 리스크가 높은 영역으로 분류됩니다.

ALIRA-AI

Workbench

2. 시간축 추적 : 인과관계 & 히스토리



- 커밋, 빌드, 배포, 이슈 해결 등 모든 이벤트를 타임라인으로 연결
- 원인-결과 인과관계 분석 (특정 커밋 <-> 이슈 해결)
- 과거 프로젝트 히스토리를 활용한 재사용/회고 지식자산화

- 이벤트 인과관계 분석, 히스토리 재사용
- 커밋-배포-이슈 이벤트 연결, 원인-결과 관계 분석
- 원인 추적: 버그의 근본 원인을 시간을 거슬러 추적

최근 프로젝트에서 가장 큰 품질 리스크가 발생한 구간은 언제야?

PALADIN

24.07 초반 auth_module 관련 커밋 3건에서 연속적으로 빌드 실패가 발생했습니다. 이는 JIRA-192와 연결되며, 인과관계 그래프 상 "인증 모듈" 노드가 리스크 허브로 표시됩니다.

이런 리스크는 어떻게 대응됐어?

PALADIN

OOO 선임이 24.07.08 commit f7a221에서 인증 로직을 수정했고, 이후 빌드 안정성이 회복되었습니다. 시스템은 동일 패턴 발생 시 사전 알람을 제공하도록 설정되었습니다.

지난달 결제 모듈 배포가 일주일 늦어진 원인이 뭐였어?

PALADIN

24.06.12 빌드에서 성능 저하 문제가 발견되었고, commit b4c911의 DB 커넥션 풀 수정이 원인이었습니다. 이로 인해 JIRA-178 이슈 해결에 추가 5일이 소요되었습니다.

그럼 최종 일정은 어떻게 조정됐어?

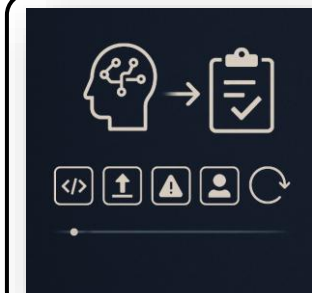
PALADIN

24.06.18에 JIRA-178이 해결되면서 06.20 배포로 조정되었고, 타임라인 상 일정 지연 요인은 기록되어 향후 회고 자료로 활용됩니다.

ALIRA-AI

Workbench

3. 자동화 액션 : 자동화



- AI 에이전트의 분석 결과를 즉시 실행
- 빌드/테스트/배포 과정의 반복 의사 결정 자동화
- 취약 코드 발견 시 자동 티켓 생성, 테스트 실패 시 담당자 자동 할당
- 새로운 티켓 등록 시 known issue 여부 판별 -> 자동 분류/할당
- DevOps 워크플로우와 직접 연결해 속도·품질 동시 확보
- 품질 관리: 코드 품질 임계치 위반시 자동 알림
- 문서 생성: 릴리즈 노트, 변경 로그, 영향도 보고서 자동 작성
- 워크플로우 / 반복 의사결정 자동화 (티켓 생성, 담당자 할당, DevOps 연동)

[중복 판정] 신규 이슈 "로그인 시 세션 만료 오류" → Known Issue

매칭: JIRA-142(과거 동일 패턴, 재발 가능성 높음)

조치: 신규 이슈 닫힘 처리, JIRA-142에 링크 및 담당자 재할당

다음 액션: 재발 방지 체크리스트 적용
<JIRA-142 열기> | <재발 방지 체크리스트>

[자동 링크]

신규 이슈(#NEW-587)가 JIRA-142와 동일 패턴으로 식별되어 NEW-587은 닫힘(duplicate) 처리되었고, 본 이슈에 연결되었습니다.

- 매칭 근거: 스택트레이스 토큰 87% 일치, 모듈=auth_session

- 권장: 회귀 테스트 스위트에 케이스 추가

[CI 실패 탐지] PaymentFlowIT.testRollback() 실패

원인 후보: DB 트랜잭션타임아웃(로그 패턴 매칭)

조치: 이슈 JIRA-312 자동 생성, 담당자 DB팀 자동 할당

다음 액션: 타임아웃=60s 제안 · 재실행 예약(30분 후)
<로그 보기 링크> | <JIRA-312 열기> | <재실행 예약>

[자동 생성] CI 실패 감지: PaymentFlowIT.testRollback()

- 원인 후보: DB timeout (pattern: "Lock wait timeout exceeded")

- 추천 액션: DB 커넥션/타임아웃 상황, 재시도 전략 적용

- 트리거: Build #842 (2025-09-12 14:10 KST)

PAD

FCTY

ONT

PIPE

CONIN

ADM

WORKSPACE / ONTOLOGY

SYSTEM ONLINE

ONTLOGY MANAGER

Schema Definition & Knowledge Graph

Entity & Relation Editor

Visual Schema Editor

Data Browser

Version History

+ Add Entity

+ Add Relation

TestCase

Test case specification

• id : string

• name : string

• status : string

validates

Requirement

Software requirement specification

• id : string

• title : string

• description : string

+2 more...

implements

SourceCode

Source code file or module

• path : string

• language : string

• lines : long

has bug

BugTicket

Bug or issue ticket

• id : string

• title : string

• severity : string

Legend

Entity

Relation

Property

Edit Entity

Name

Requirement

Type

entity

Description

Software requirement specification

☐ Abstract

Properties

id

string

☒ required

title

string

☒ required

description

string

☐ required

priority

string

☐ required

21

PAD

FCTY

ONT

PIPE

CONN

ADM

SET

WORKSPACE / PIPELINE

PIPELINE STUDIO

Manage Data Ingestion Flows

파이프라인

Processor 확장

PIPELINES

Legacy Docs Import

SMB 공유 폴더의 레거시 문서를 수집

SMB

data-ingestion ingestion

Never

Ontology Event Monitor

온톨로지 변경 이벤트를 감지하여 관련 파이프라인 트리거

TypeDB

event-monitoring event-driven

5s ago

External Webhook Listener

외부 시스템의 웹훅 이벤트를 수신하여 처리

Webhook

event-monitoring event-driven

1m ago

RAG Document Pipeline

PDF 문서를 처리하여 RAG 시스템 구축 및 온톨로지 자동 매핑

PDF

rag-pipeline rag

10m ago

Smart Ontology Mapper

다중 소스 데이터를 온톨로지에 지능형 매핑

Multi-Source

ontology-mapping mapping

Never

EXECUTION FLOW

EVENT_LISTENER

Webhook Receiver

PROCESS

Payload Validator

PROCESS

LLM Event Classifier

TARGET

Ontology Update

CONFIGURATION

Webhook Receiver

ENDPOINT URL

admin@jerry.ai

이벤트 타입

온톨로지 변경 (Ontology Change)

소스 연결

TypeDB Connection

이벤트 필터 (JSON)

```
{ "entity_type": "Requirement", "operation": "create" }
```

DATA PREVIEW

Valid JSON

```
{  "webhook_id": "wh-456",  "source": "external-system",  "payload":    {      "event": "deployment_completed",      "environment": "production",      "version": "v1.2.3",      "timestamp": "2024-01-15T10:40:00Z"    }  }
```

WORKSPACE / PIPELINE

SYSTEM ONLINE

PROCESSOR EXTENSIONS

확장 가능한 프로세서 개발 및 관리

파이프라인

Processor 확장

CATEGORIES

전체

변환

필터링

풍부화

검증

커스텀

전체 프로세서

프로세서 검색...

v1.0.0

Document Reader

다양한 문서 형식(PDF, DOCX, MD)을 읽고 텍스트로 변환

System transformation

v1.0.0

Simulink Model Reader

Simulink 모델 파일을 읽고 구조를 추출

System transformation

v1.0.0

Code Reader

소스 코드 파일을 읽고 AST로 파싱

System transformation

v1.0.0

Document Chunker

문서를 의미 단위로 청킹

System transformation

v1.0.0

OCR Processor

이미지에서 텍스트를 추출하는 OCR 처리

System transformation

v1.0.0

Image Processor

이미지 파일을 처리하고 메타데이터 추출

System transformation

v1.0.0

Video Processor

비디오 파일에서 프레임 추출 및 분석

System transformation

v1.0.0

Audio Processor

오디오 파일을 처리하고 텍스트로 변환

System transformation

v1.0.0

JSON Parser

JSON 데이터를 파싱하고 구조화

System transformation

+ 새 프로세서 등록

24

WORKSPACE / FACTORY

SYSTEM ONLINE

← 인벤토리로 돌아가기

템플릿 선택 > 기본 설정 > 온톨로지 연결 > UI 구성 > AI 모델 > 배포

PAD

FCTV

ONT

PIPE

CONN

ADM

앱 템플릿 선택

사전 구성된 AI 응용 프로그램 템플릿을 선택하거나 빈 프로젝트에서 시작하세요

템플릿 검색...

전체

AI Assistant

Analytics

Automation

Data Processing

Monitoring

AI Knowledge Assistant

RAG 기반 지식 검색 및 Q&A 챗봇. 온톨로지 데이터를 활용한 맥락 기반 답변 생성

RAG LLM ChatBot

intermediate • 15분

UI

Analytics Dashboard

실시간 데이터 시각화 및 분석 대시보드. 커스텀 차트와 KPI 모니터링

Dashboard Charts Analytics

beginner • 10분

UI

Data Ingestion Agent

자동화된 데이터 수집 및 처리 에이전트. 스케줄링과 변환 파이프라인 포함

ETL Automation Pipeline

advanced • 20분

AGENT

Semantic Search Engine

벡터 기반 의미론적 검색 엔진. 자연어 쿼리로 관련 문서 검색

Search Embeddings NLP

intermediate • 15분

UI

Workflow Automation

노코드 워크플로우 자동화 플랫폼. 드래그 앤 드롭으로 프로세스 구성

Workflow Automation No-Code

intermediate • 12분

UI

Monitoring & Alerts

시스템 모니터링 및 실시간 알림. 임계값 기반 자동 알림 발송

Monitoring Alerts Metrics

advanced • 18분

AGENT

이전

단계 1 / 6

다음

25

WORKSPACE / FACTORY

SYSTEM ONLINE

← 인벤토리로 돌아가기

Developer Hub

PSDK 다운로드, 온톨로지 ORM 사용법, API 문서

Overview

Quick Start

Examples

API Reference

⚡

Live API Documentation

LIVE

현재 온톨로지를 기반으로 실시간 생성된 API 문서

4

Entities

3

Relations

14

Properties

⚡ Live API 문서 열기

📦

Paladin SDK (PSDK) 다운로드

PSDK는 온톨로지 ORM, 앱 구조, 빌드 시스템, 단위 테스트를 포함한 완전한 개발 키트입니다.

📄

Python SDK

TypeDB ORM with type-bridge

v1.2.3 • Python 3.13+

다운로드 (5.2 MB)

📄

TypeScript SDK

React + TypeDB Driver

v2.0.1 • Node 18+

다운로드 (12.8 MB)

패키지 내용

> 온톨로지 ORM (type-bridge 기반)

26

ALIRA-AI Docs

New
Docs

ALIRA-AI docs

새 문서 만들기

🔍 템플릿 검색...

전체 (20)

요구사항 정의서 (4)

설계 문서 (1)

구현 문서 (1)

애자일 개발 (3)

자동차 도메인 (3)

국방/신뢰성 (2)

간소화 문서 (3)

일반 문서 (3)

모든 템플릿 (20개)



시스템 요구사항 명세서 (SRS)

요구사항 정의서



사용자 스토리

애자일 개발



회의록

일반 문서



기능 안전성 (ISO 26262)

자동차 도메인



신뢰성 시험 계획서

국방/신뢰성



간소화 설계 문서

간소화 문서



시스템 아키텍처 설계서

설계 문서



ASPICE 컴플라이언스

자동차 도메인



회고 (Retrospective)

애자일 개발



V-Cycle 개발 문서

자동차 도메인



프로젝트 제안서

일반 문서



간소화 테스트 계획서

간소화 문서



코딩 표준 및 가이드라인

구현 문서



간소화 요구사항 문서

간소화 문서



스프린트 계획

애자일 개발



MIL-STD 기반 문서

국방/신뢰성



기술 보고서

일반 문서



아키텍처 요구사항 명세서

요구사항 정의서

시스템 아키텍처 수준의 요구사항을 정의하는 상세 문

총 20개 중 20개 표시

New
Docs

ALIRA-AI docs

파일 편집 보기 도움말

EXPLORER

요구사항 정의서 3

기능 안전성 (ISO 26262)

단위/모듈 요구사항 명세서

시스템 요구사항 명세서 (SRS)

설계 문서 0

구현 문서 1

코딩 표준 및 가이드라인

테스트 문서 0

배포 문서 0

ISO 26262 기능 안전 문서

저장됨 오후 9:41:12

프로젝트 정보

- 프로젝트명: bluez comm 기능
- ASIL 등급: ASIL B
- 시스템/아이템: 자율주행 브레이크 시스템 (AEB)
- ISO 26262 적용 버전: ISO 26262:2018
- 작성일: 2024년 5월 10일
- 작성자: 이수진 (기능 안전 엔지니어)

1. 개요 및 범위

개요

스마트 분석을 수행하고 있어요
AI가 마법을 부리는 중...

batterydump.c C 326줄

54줄 선택됨

```
182 {
183     int ret = 0;
184     struct battery_monitor_current_s current;
185
186     ret = ioctl(fd, BATIOC_STATE, (unsigned long)&status);
187     if (ret < 0)
188     {
189         printf("BATIOC_STATE failed %d", ret);
190     }
191
192     ret = ioctl(fd, BATIOC_ONLINE, (unsigned long)&online);
193     if (ret < 0)
194     {
195         printf("BATIOC_ONLINE failed %d", ret);
196     }
197
198     ret = ioctl(fd, BATIOC_HEALTH, (unsigned long)&health);
199     if (ret < 0)
200     {
201         printf("BATIOC_HEALTH failed %d", ret);
202     }
203
204     ret = ioctl(fd, BATIOC_VOLTAGE, (unsigned long)&vol);
205     if (ret < 0)
206     {
207         printf("BATIOC_VOLTAGE failed %d", ret);
208     }
209
210     ret = ioctl(fd, BATIOC_COULOMBS, (unsigned long)&coulomb);
211     if (ret < 0)
212     {
213         printf("BATIOC_COULOMBS failed %d", ret);
214     }
215
216     memset(&current, 0, sizeof(struct battery_monitor_current_s));
217     ret = ioctl(fd, BATIOC_CURRENT, (unsigned long)&current);
218     if (ret < 0)
219     {
220         printf("BATIOC_CURRENT failed %d", ret);
221     }
222
223     printf("mask:%PRIx32, state:%d, online:%d, vol:%d, coulombs:%d, "
224           "current:%PRIi32, measurement time:%PRIu32\n",
225           mask, status, online, vol, coulomb, current.current,
226           current.time);
227     return;
228 }
229
230 static void dump_battery(int fd, int chiptype, uint32_t mask)
231 {
232     switch (chiptype)
233     {
234     case CHIP_TYPE_CHARGER:
235         dump_charger(fd, mask);
236         break;
237     case CHIP_TYPE_GAUGE:
238         dump_gauge(fd, mask);
239         break;
240     case CHIP_TYPE_MONITOR:
241         dump_monitor(fd, mask);
242         break;
243     }
```

AI add-smart 작업을 시작합니다... 기능 안전성 (ISO 26262)

© 2024 ALIRA AI Docs Desktop

v1.0.0

New
Docs

The screenshot shows the ALIRA-AI Docs Desktop application. The interface includes a top bar with the title 'ALIRA-AI docs' and a right sidebar with icons for file, edit, view, and search. The main area is divided into three panels:

- EXPLORER:** A sidebar on the left showing a file tree. The selected item is '기능 안전성 (ISO 26262)'. Other items include '요구사항 정의서', '단위/모듈 요구사항 명세서', '시스템 요구사항 명세서 (SRS)', '설계 문서', '구현 문서', '코딩 표준 및 가이드라인', '테스트 문서', and '배포 문서'.
- Document Content:** The central panel displays the document 'ISO 26262 기능 안전 문서'. It includes a '프로젝트 정보' (Project Information) section with details like project name, ASIL level, system, version, creation date, and author. Below this is a '1. 개요 및 범위' (Overview and Scope) section with a '개요' (Overview) subsection. The text describes the document's purpose in implementing battery monitoring for the ASIL B system and references the ISO 26262:2018 standard.
- Code Editor:** The right panel shows the source code for 'batterydump.c'. The code is in C and implements functions for monitoring battery status (current, voltage, health, coulombs) and dumping this information. It uses ioctl system calls to interact with hardware and includes error handling with printf statements.

At the bottom of the window, there is a status bar showing the current document '기능 안전성 (ISO 26262)', the copyright '© 2024 ALIRA AI Docs Desktop', and the version 'v1.0.0'.

Creating with ALiRA-AI, Validating with VERIFAI The Complete Solution

Test by AI

ALIRA-AITM Powered

ALIRA-AI Hub
SW 공학 AI 모델 컬렉션

ALIRA-AI Workbench
Agentic AI 솔루션

ALIRA-AI Docs
Code-AI-Doc 솔루션

ALIRA-AI Workspace
End-to-End Agentic Automation

Test of AI

VERIFAITM Solutions

VERIFAI-M
AI 모델에 대한 검증 솔루션

VERIFAI-A
Agentic AI 검증 솔루션

VERIFAI-D
AI 시스템 검증을 위한 데이터 솔루션

VERIFAI-V
비전 모델 데이터 증강 솔루션