

보안 과제(), 일반 과제() / 공개(), 비공개() 발간등록번호()
국토교통기술촉진연구사업 제2차 연도 최종보고서

R&D / 제48호 - 1122

SOC 시설물의 스마트 통합 유지 관리를 위한 드론 기반 3D 엔진 및 무선 제어 계측 플랫폼 기반 기술 개발 최종보고서

2019.01.18.

주관연구기관 / 대림대학교산학협력단
협동연구기관 / (주)케이지아이

국토교통부
국토교통과학기술진흥원

제 출 문

국토교통과학기술진흥원장 귀하

‘SOC 시설물의 스마트 통합 유지 관리를 위한 드론 기반 3D 엔진 및 무선 제어 계측 플랫폼 기반 기술 개발’(연구개발 기간 : 2017.04.01 ~ 2018.12.31) 과제의 최종보고서 10부를 제출합니다.

2019. 01. 18.

주관연구기관명 : 대림대학교산학협력단 (대표자) 박상운 (인)

협동연구기관명 : (주)케이지아이 (대표자) 김종훈 (인)

주관연구기관책임자: 이영걸

협동연구기관책임자: 이윤래

국토교통부소관 연구개발사업 운영규정 제29조에 따라 최종보고서 열람에
동의합니다.

보고서 요약서

과제 고유 번호	제48호 - 1122	해당 단계 연구 기간	2017. 04. 01 - 2018. 12. 31(21개월)	단계구분	해당없음
연구사업명	중사업명	국토교통기술촉진연구사업			
	세부사업명				
연구과제명	대과제명	대과제가 있을 경우 기재합니다(단위과제일 경우에는 아래에 기재합니다)			
	세부과제명	SOC 시설물의 스마트 통합 유지 관리를 위한 드론 기반 3D 엔진 및 무선 제어 계측 플랫폼 기반 기술 개발			
연구책임자	이영걸	해당단계 참여연구원 수	총: 명 내부: 명 외부: 명	해당단계 연구비	정부: 천원 민간: 천원 정부외: 천원 상대국: 천원 계: 천원
		총 연구기간 참여연구원 수	총: 14명 내부: 14명 외부: 명	총 연구비	정부: 370,000천원 민간: 125,000천원 정부외: 천원 상대국: 천원 계: 495,000천원
연구기관명 및 소속 부서명	대림대학교산학협력단			참여기업명 (주)케이지아이	
국제공동연구	상대국명:			상대국 연구기관명:	
위탁연구	연구기관명:			연구책임자:	

※ 국내·외의 기술개발 현황은 연구개발계획서에 기재한 내용으로 같음

연구개발성과의 보안등급 및 사유	
-------------------------	--

9대 성과 등록·기탁번호

구분	논문	특허	보고서 원문	연구시설 · 장비	기술요약 정보	소프트 웨어	화합물	생명자원		신품종	
								생명 정보	생물 자원	정보	실물
등록·기 탁 번호											

국가과학기술종합정보시스템에 등록된 연구시설·장비 현황

구입기관	연구시설 · 장비명	규격 (모델명)	수량	구입연월일	구입가격 (천원)	구입처 (전화)	비고 (설치장소)	NTIS 등록번호

SOC 시설물의 스마트 통합 유지 관리를 위한 드론 기반 3D 엔진 및 무선 제어 계측 플랫폼 기반 개발, 정확하고 효율적인 시설물 유지 관리를 위해 현장용 계측 장비와 연계하여 시설물 내 입체적 위치관리가 가능한 3차원 시설물 모형의 자동 구축을 가능하게 지원하는 SW 엔진 모듈과 재난 재해에 대비한 유지 관리용 감시 시스템에 IoT 기술을 접목하여 스마트폰과 연동이 가능한 인터페이스 및 3D 시설물 관리를 위한 스마트 유지관리 통합 플랫폼 기반 기술 개발	보고서 면수 132
---	--

〈 국문요약문 〉

연구의 목적 및 내용	정확하고 효율적인 시설물 유지 관리를 위해 현장용 계측 장비와 연계하여 시설물 내 입체적 위치관리가 가능한 3차원 시설물 모형의 자동 구축을 가능하게 지원하는 SW 엔진 모듈과 재난 재해에 대비한 유지 관리용 감시 시스템에 IoT 기술을 접목하여 스마트폰과 연동이 가능한 인터페이스 및 3D 시설물 관리를 위한 스마트 유지관리 통합 플랫폼 기반 기술 • 라이다 카메라를 장착한 드론을 이용하여 대상 시설물 촬영 및 위치 기반 3D 모델링 모듈 개발 • 시설물 내 센서 정보의 입체적 위치관리가 가능한 시설물 3D 가상 모형 구축 모듈 개발 • 계측 정보 데이터베이스와 연결이 가능한 3D 시설물 뷰어 모듈 개발 • 현장용 무선통신 기반 계측센서 다중 인터페이스 연동 및 스마트폰용 통신 관리 앱 시스템 모듈 개발 • 재난·재해용 통합 정보 플랫폼과의 효과적인 연계를 위한 인터페이스 구현 모듈 개발				
연구개발성과	SOC 시설물 형상을 이용하여 스마트 통합 유지 관리를 위한 드론 기반 Automatic 3D 엔진 모듈 및 IoT 기술을 활용한 Non-Stop 무선 제어 계측 센서 연동 통합 플랫폼 기반 기술 개발 • SOC 시설물 관리 통합 프레임워크 기반 기술 개발 • SOC 시설물 관리 통합 소프트웨어 컴포넌트 개발 • SOC 시설물 관리 통합 애플리케이션 개발				
연구개발성과의 활용계획 (기대효과)	- 실시간 감시가 필요한 인명이나 재산 피해가 예상되는 재난 예상 지역에 대한 현황 및 위험 예측 분야에 활용 (예) 사면, 도로, 지진, 산불 등 - 2015년 국내 건설-IT 시장은 59조원, 2020년에는 세계 건설-IT 시장의 규모가 4,500억 달러에 이를 정도로 거대 시장으로 확대될 것으로 전망 - 통합 정보를 기반으로 현황, 이력, 피해 정의, 피해 분석기술을 근거로 복합 재난·재해에 효과적으로 대응, 확산 차단 등 피해 최소화를 통해 재난 복구비용 절감 가능				
핵심어 (5개 이내)	사회간접자본	사물인터넷	계측센서	드론	3D 엔진

＜ SUMMARY ＞

Purpose & Contents	For accurate and efficient maintenance of social facilities, as a total smart software solution system, on-site measurement equipment information is integrated to 3-dimensional shape of facilities automatically. It is also equipped with IoT technology, it enables management through smart devices. • Development of snapshot acquisition and 3-dimensional modeling modules using drone equipped with LIDAR Camera. • Development of integration modules with 3-dimensional data and on-site digital sensor data of target facility • Development of 3-D facility viewer modules with status monitored data • Development of multiple interface modules for wireless communication among sensors and communication configuration app for smart device • Development of integrated information platform of disaster management for heterogeneous data source for other facilities				
Results	Development of drone-equipped 3D Engine and wireless control measurement platform technology for Integrated management of SOC facilities				
Expected Contribution	– Disaster management application for hazardous area, status monitoring, alarm generation and risk analysis, especially required realtime response. (Ex.) Landslide, Road, Earthquake, Wildfire etc. – Construction and Civil engineering with IT technology market is growing rapidly and global market is also growing and estimated approximately 4.5 billion(USD). – Economic and social cost of disaster will be reduced by realtime integrated information management system with situation monitoring, early warning, decision support, risk prediction, situation analysis, cause analysis and damage Assessment.				
Keywords	SOC	IoT	Sensor	Drone	3D Engine

< 목 차 >

1. 연구개발과제의 개요	01
2. 국내외 기술 개발 현황	09
3. 연구수행내용 및 성과	12
4. 목표 달성도 및 관련 분야 기여도	116
5. 연구개발성과의 활용 계획	125
6. 연구 과정에서 수집한 해외 과학기술 정보	125
7. 연구개발성과의 보안등급	125
8. 국가과학기술종합정보시스템에 등록한 연구시설·장비 현황	126
9. 연구개발과제 수행에 따른 연구실 등의 안전 조치 이행 실적	126
10. 연구개발과제의 대표적 연구 실적	126
11. 기타 사항	126
12. 참고 문헌	127

제1장. 연구개발과제의 개요

1. 연구개발 목적

(1) 최종 목표

- 정확하고 효율적인 시설물 유지 관리를 위해 현장용 계측 장비와 연계하여 시설물 내 입체적 위치관리가 가능한 3차원 시설물 모형의 자동 구축을 가능하게 지원하는 SW 엔진 모듈
- 지진, 수재해 등 재난·재해에 대비한 유지 관리용 감시 시스템(영상 정보, 계측장비 등)에 IoT 기술을 접목하여 스마트폰과 연동이 가능한 인터페이스 및 3D 시설물 관리를 위한 스마트 유지관리 통합 플랫폼 기반 기술

(2) 세부 목표

① SOC 시설물 형상을 이용하여 스마트 통합 유지 관리를 위한 드론 기반 Automatic 3D 엔진 모듈 개발

- 라이다 카메라를 장착한 드론을 이용하여 대상 시설물 촬영 및 영상정보 수집
- 스마트 유지관리 기반을 위한 시설물 3D 속성정보와 연계된 시범 DB 설계 및 구축
- 위치(GPS) 및 촬영 정보를 이용한 위치 기반 3D 모델링 모듈 개발
- 시설물 내 센서 정보의 입체적 위치관리가 가능한 시설물 3D 가상 모형 구축 모듈 개발
- 계측 정보 데이터베이스와 연결이 가능한 3D 시설물 뷰어 모듈 개발

② IoT(Internet of Thing, 사물인터넷) 기술을 활용한 Non-Stop 무선 제어 계측 센서 연동 통합 플랫폼 기반 기술 개발

- 현장용 무선통신 기반 계측센서 다중 인터페이스 연동 모듈 개발
- 무선(블루투스, WiFi/LTE) 네트워크를 이용하여 계측기와 연동이 가능한 스마트폰용 통신 관리 앱 모듈 개발
- 수집된 계측 정보의 정확한 입체적 관리를 위한 시설물 3D 정보 연동 인터페이스 모듈 구현
- 재난·재해(지진 및 수재해 등)용 통합 정보 플랫폼과의 효과적인 연계를 위한 (시설물 및 감시 정보) 인터페이스 모듈 구현
- 실시간 재난 정보의 3D 시설물 표출 시스템 시제품 제작

(3) 2차년도 목표

① 주관연구기관(대림대학교 산학협력단):

- 현장용 무선통신 기반 계측센서 다중 인터페이스 연동 모듈 개발
- 수집된 계측 정보의 정확한 입체적 관리를 위한 시설물 3D 정보 연동 인터페이스 모듈 구현

② 참여기관 (㈜케이지아이):

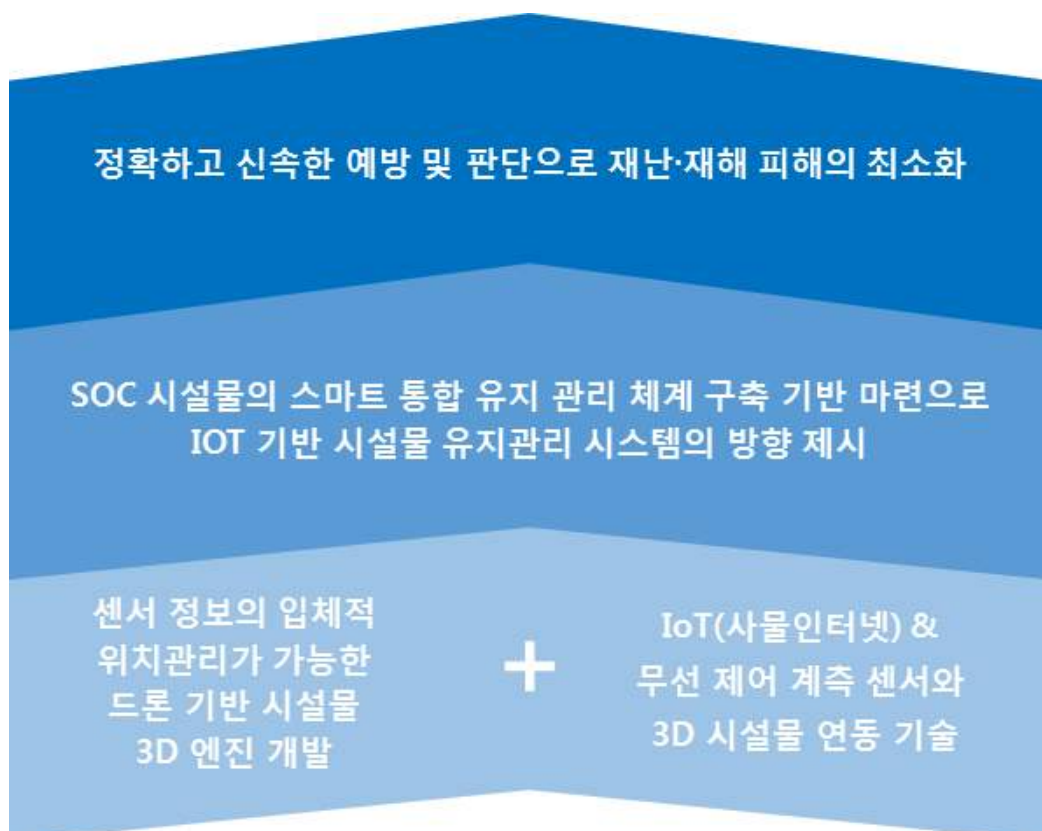
- 무선(블루투스 , WiFi/LTE) 네트워크를 이용하여 계측기와 연동이 가능한 스마트폰용 통신 관리 앱 모듈 개발
- 재난·재해(지진 및 수재해 등)용 통합 정보 플랫폼과의 효과적인 연계를 위한 (시설물 및 감시 정보) 인터페이스 모듈 구현
- 실시간 재난 정보의 3D 시설물 표출 시스템 시제품 제작

2. 연구개발의 필요성

- 2016년 9월 12일 경주 지진(규모 5.8) 등 최근 대규모의 지진 발생이 점차 증가하는 추세에 있으며, 지진 발생 시 광범위한 지역에 걸쳐서 막대한 인명피해와 경제적 손실을 일으키며, 국가 신경망의 마비 등 국가적 재난상황을 초래함.
- 재난의 대형화, 신종·복합화로 사회 전반의 재난위험이 증가하고 있으며 피해 규모도 대형화되어 재난·안전 분야에 대한 국민적 관심도가 매우 높아지는 실정임.



- 국가와 지방자치단체는 재난이나 그 밖의 각종 사고로부터 국민보호, 재난이나 각종 사고 예방, 발생한 피해를 신속 대응·복구하기 위해 ‘재난 및 안전기술개발 종합계획’을 수립·시행하고 있으나 데이터 표준화가 미흡하고 신규 비정형 데이터 등에 대한 상호연동체계가 부족하여 체계적인 관리와 활용을 위한 통합관리체계가 미흡함.
- 급증하는 각종 재난별 대응전략을 수집하고 현안 해결을 지원하기 위하여 각종 정형/비정형 재난 데이터들을 수집하여 분석할 수 있는 연동 기술 개발이 요구됨.
- 현재 재난재해 대응 ICT 기술은 시설물 설비 분야에서 가장 낙후되고 폐쇄적이라고 평가되고 있으며, 이는 법규 중심의 수동성과 재난·재해 관리시스템 공급자의 기술 이기주의 등으로 인하여, 재난·재해 관리시스템의 개방성이 떨어지고 국가 R&D 등에 의해 개발된 신기술의 현장적용이 미흡한 실정임.
- 국내에 구축되어 있는 유무선 통신기술 및 정보기술(IT)은 이미 세계적 수준으로 이를 이용한 지능형 교통체계, 스마트 도로, 원격의료 등 다양한 연구가 수행되고 있으나, 재난·재해를 위한 통신 인프라를 운용하여 지진·화재·침수 및 복합재난관리를 위한 콘텐츠 또는 솔루션의 개발 노력이 아직까지는 미흡한 실정임.
- 다양한 SOC 시설물 관리를 위해서는 단순 시설물 정보는 물론 부재단위 수준의 세밀한 기반 정보가 필요하며, 이는 **3D 모형 형상 정보**와 **센서 정보**가 결합된 **복합 가상 공간정보 기술**을 통해 해결할 수 있을 것으로 판단됨.
- **정확하고 효율적인 시설물 유지 관리**를 위해 **현장용 계측 장비와 연계**하여 시설물 내 입체적 위치관리가 가능하도록 드론을 활용하여 **3차원 시설물 모형의 자동 구축**을 하고, 이를 지진, 수재해 등 **재난 재해에 대비한 유지 관리용 감시 시스템**(영상 정보, 계측장비 등)에 **IoT 기술을 접목**하여 스마트폰과 연동이 가능한 인터페이스 및 3D 시설물 관리를 위한 **스마트 유지관리 통합 플랫폼 기반 기술**을 개발하는 것을 목표로 함.

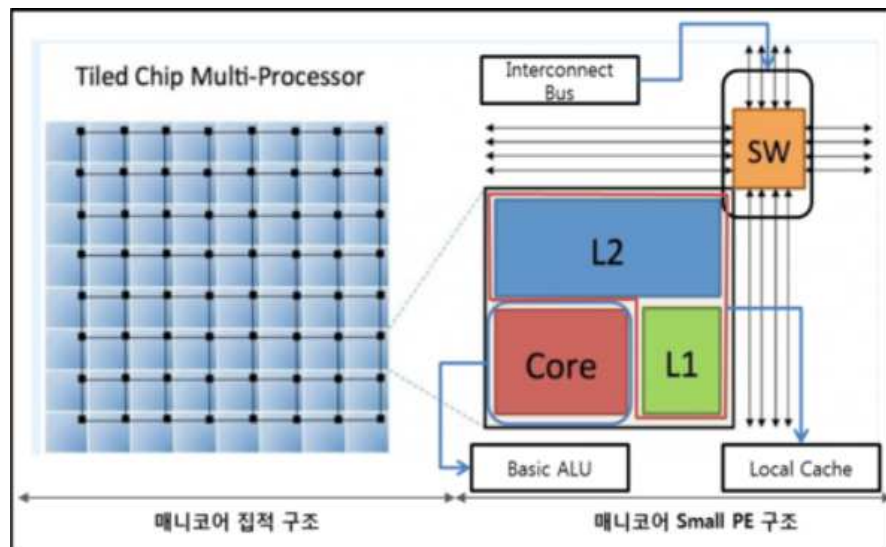


3. 연구개발 범위

- 주관연구기관(대림대학교 산학협력단):

o 현장용 무선통신 기반 계측센서 다중 인터페이스 연동 모듈 개발

- 대형 센서 네트워크에서 발생하는 방대한 양의 데이터를 효율적으로 처리하고 전송하기 위한 다중 인터페이스 알고리즘 개발
 - 구조물의 정밀한 상태평가를 위해 센서 네트워크의 센서 수도 증가하게 되며, 이러한 대형 센서 네트워크에서 발생하는 방대한 양의 데이터를 효율적으로 처리하고 전송하기 위한 알고리즘 개발이 필수적임
 - 많은 양의 데이터를 처리하기 위해서는 병렬처리(Parallel processing) 알고리즘 도입 검토가 필요함
 - 다수의 센서에서 발생하는 데이터 처리를 위한 다중 처리 알고리즘 구현



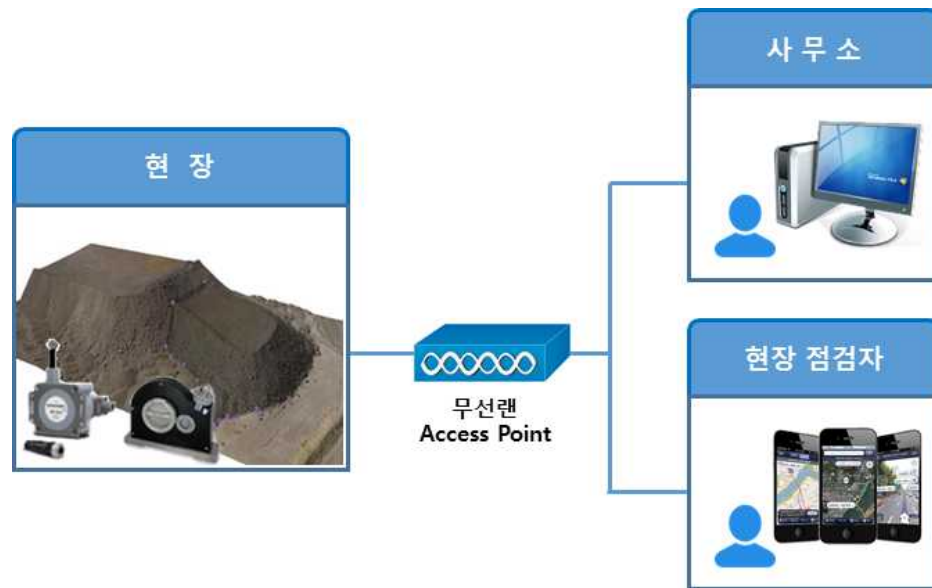
[병렬처리를 위한 PE(처리 요소, Processing Element)구성 방법]

- 센서 네트워크의 데이터 처리 알고리즘

절차 1. 센서 네트워크에서 발생하는 방대한 데이터를 효율적으로 처리하여 전송하기 위한 알고리즘 및 분산 프로세스 구현

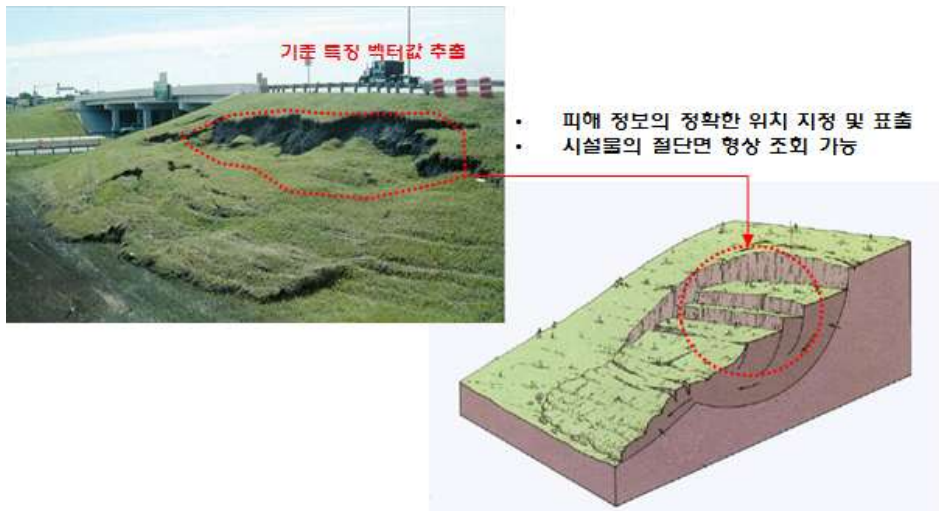
절차 2. 네트워크 상의 분산 컴퓨팅(distributed computing)에 적합한 데이터 처리 및 피로누적(fatigue accumulation) 추정 알고리즘 연구

- 계측기와 연결되는 기기와의 연결 인터페이스를 구축



[네트워크를 통한 실시간 전송기술]

- o 수집된 계측 정보의 정확한 입체적 관리를 위한 시설물 3D 정보 연동 인터페이스 모듈 구현
 - 시설물 3D의 입체적 관리를 위한 체계적인 데이터 표현 루틴 연구 및 개발
 - 국내에서 정부기관에서 제공하는 자연재해 정보는 서로 달라 수집된 계측정보는 상황인식을 위해 매우 제한적으로 사용됨
 - 대상 시설물의 기록 표준화를 통해 체계적인 데이터 표현이 필요함
 - 대상 시설물의 이상 징후 관리를 위한 시설물 정보의 연동 DB 설계
 - 기준 영상 및 3D 깊이 정보
 - 입력 영상 및 3D 깊이 정보
 - 기준 특징벡터 $X_r = \{X_{r_1}(x_{r_1}, y_{r_1}, d_{r_1}), X_{r_2}(x_{r_2}, y_{r_2}, d_{r_2}), \dots, X_{r_N}(x_{r_n}, y_{r_n}, d_{r_n})\}$
 - 입력 특징벡터 $X_i = \{X_{i_1}(x_{i_1}, y_{i_1}, d_{i_1}), X_{i_2}(x_{i_2}, y_{i_2}, d_{i_2}), \dots, X_{i_N}(x_{i_n}, y_{i_n}, d_{i_n})\}$
 - 지반 및 지반구조물의 최적의 변형 또는 진동 매트릭스 정보 관리를 위한 계측장비 위치 및 속성 저장 구조 설계 실시
 - 재난재해 감시를 위한 자동 계측 정보를 연계할 수 있는 인터페이스 개발
 - 연계된 정보를 1차년도에 구축한 3D 시설물 정보 내 Display 및 Control Module 개발



[계측 정보와 시설물 3D 정보 연동 화면(예시)]

- 참여기관 (㈜케이지아이):

- o 무선(블루투스 , WiFi/LTE) 네트워크를 이용하여 계측기와 연동이 가능한 스마트폰용 통신 관리 앱 모듈 개발
 - 유선 네트워크를 이용한 센서에 기반한 모니터링 시스템은 센서 및 케이블 설치에 따르는 막대한 비용 때문에, 대형 구조물의 상태평가 및 모니터링에 필요한 고밀도 센서 시스템에 적합하지 않다는 태생적 한계를 가짐
 - 유선 네트워크가 가진 단점을 극복하기 위해 무선 스마트 센서를 통해 데이터를 송수신 및 제어가 가능한 앱 시스템 개발
 - 블루투스 및 WiFi/LTE를 이용한 무선 모듈을 적용한 현장용 계측기 개발
 - 계측 센서 연동 및 무선통신을 이용한 데이터 송수신 모듈 개발
 - 하드웨어와 소프트웨어(앱 시스템)와의 통신 프로토콜 정의 및 모듈 구현
 - 측정된 정보를 수작업이 아닌 앱 프로그램을 통한 자동 입력

계측기와 연동한 현장용 앱 시스템	▪ 서버 OS : Windows / Linux 계열 ▪ 스마트폰 OS : 안드로이드 5.0(블리썬) 이상 ▪ 데이터베이스 : MS-SQL 2008 ▪ 개발 TOOL : JAVA(JDK 1.7.2(Update 18), XML ,안드로이드 SDK ▪ 통신 프로토콜 : HTTP, TCP/IP 용
-----------------------------------	--

[구축 시스템 사양]

[진동 측정 앱 시스템 예]

o 재난·재해(지진 및 수재해 등)용 통합 정보 플랫폼과의 효과적인 연계를 위한 시설물 감시 정보 인터페이스 모듈 구현

- FMS(시설물 정보관리 종합정보 시스템)

- 시설물의 안전관리에 관한 특별법에 의한 국가 주요 시설물(1,2종 시설물)의 안전 및 유지관리에 관련된 정보체계를 구축하기 위하여 시설물의 정보, 안전진단전문기관 및 유지관리업자의 정보 등을 종합 관리하는 시스템
- 시설물 관리대장, 유지관리계획/실적보고, 시설물비용정보, 점검진단대가산정, 설계도서(보고서), 관련업체(안전진단전문기관, 시설물유지관리업체)정보관리, 실적확인서 발급, 기술정보(시설물사고사례, 유지관리기술정보), 현황 및 통계, 커뮤니티 등의 기능으로 구성
- 대상시설물 : 교량, 터널, 항만, 댐, 하천(수문, 제방, 하구둑, 방조제, 보), 상수도, 공공하수처리장, 폐기물매립시설, 건축물, 옹벽, 절토사면 등



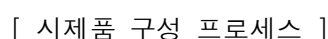
[FMS 데이터 정보 구성도]

- o 실시간 재난 정보의 3D 시설물 표출 시스템 시제품 제작
 - 다음과 같은 과정을 통해서 시제품을 제작함
 - 본 연구에서는 연구된 요소 기술을 종합적으로 통합 연계하여 운용되는 과정을 구성함



- 계측 정보 연계를 위한 데이터 표준 정의 및 인터페이스 설계
- 실시간 계측 센서 정보를 취합하여 공유를 지원할 수 있는 인터페이스 설계
- 통합 정보 플랫폼과의 효과적인 연계를 위한 시설물 감시 정보 인터페이스 구현

- 다음과 같은 과정을 통해서 시제품을 제작함
- 본 연구에서는 연구된 요소 기술을 종합적으로 통합 연계하여 운용되는 과정을 구성함



제2장. 국내외 기술 개발 현황

○ 국내 현황

• 국내 기술 현황

- 시설물 관리에 대한 관심도가 매우 높으며, 원격 시설물 모니터링 분야에 대해 개념적이고, 실험적인 연구가 지속적으로 진행되어 왔음.
- 대부분의 연구가 특정 장치 시설물에 대한 이력과 현황에 대해 분절화된 형태로 추진되어 왔음. 특히 실시간 모니터링 부분에 대해서는 통신비용 문제와 성능의 한계로 인해 완성된 체계를 실현하지 못하고 있음.
- 선행적인 조치가 필요한 시설물 모니터링 분야는 단편적 현황 점검으로는 적극적으로 활용하기 어려운 상황이며, 이에 대한 데이터의 공유 역시 이루어지지 않아 복합적인 사고와 재난으로 확대될 가능성이 매우 높음.
- 특히 특정 센서 장치와의 단순 통신을 통한 연구가 주로 진행되어 전체 상황에 대한 선행 조치, 현황, 사후 원인 분석 등 복잡한 재난 관리에 활용이 불가능하였음.

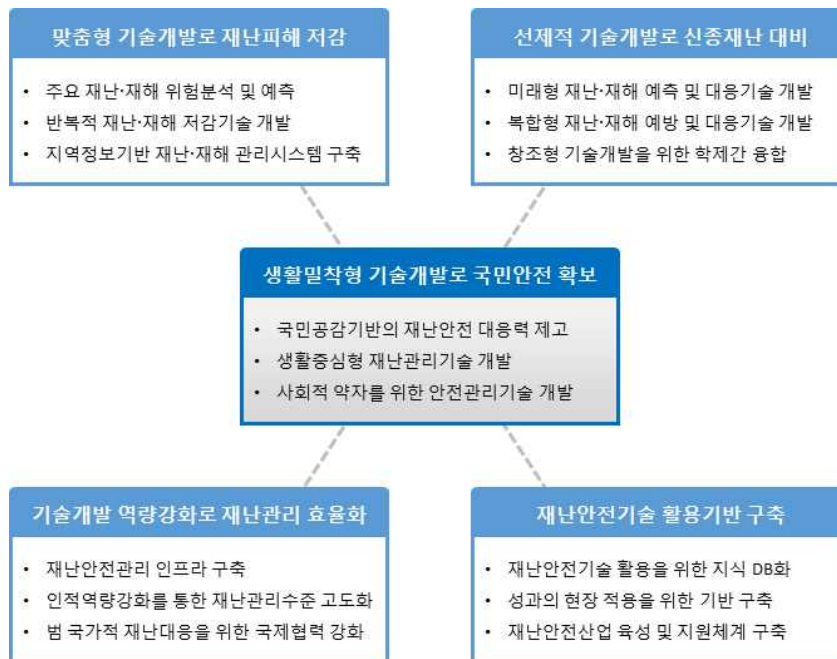
• 국내 표준화 현황

- 국내에서는 소방방재청, 기술표준원, 한국표준협회, 한국정보통신기술협회(TTA)를 중심으로 재난관련 비상통신서비스에 대한 표준화 활동을 추진함.
- 재난안전관리와 관련한 5건의 ISO 국제표준에 대해 2건이 한국산업표준(KS)로 제정 고시됨.
- 미래부는 재난안전플랫폼 기술개발 계획에 따라 지리위치정보, 기상정보, 소방정보 등 각종 재해 재난 수치 정보의 연계를 위한 국내 표준화 추진 예정임.
- 각종 재난감시 정보의 효과적 활용을 위해 재난감시정보의 수집, 분석, 가시화 기술개발 및 국내 표준화도 함께 추진 예정임.
- 국가 재난관리 추진체계상 재난관리 책임기관에서 제한된 목적으로 일부 재난관리 표준화를 수립하여 각각 별도로 재난관리 업무에 사용하고 있으므로, 통합적 재난관련 운영(재난/방재 서비스 모델, 추진 방법 및 가이드라인)에서의 표준화를 진행해야 함.
- 국민안전처는 국가재난관리정보시스템(NDMS)를 구축하여 '15년 기준 52개 기관 260종의 재난정보를 통합, 관리하고 단일화된 상황전파체계를 제공하고 있으나 데이터 표준화 및 신규 비정형데이터 등에 대한 상호연동체계가 미흡함.

○ 국외 현황

• 국외 기술 현황

- 국내 재난·재해 분야 기술수준은 점차 개선되고 있으나 미국 대비 약 6년의 기술격차를 보여 기초연구 및 응용개발 등의 지속적인 개선 필요함.
- 선진국의 경우 Hazards Data Distribution System(HDDS) 등 지속적인 표준화된 복합재난 데이터 기반의 분석 플랫폼 연구개발을 통해 지진 및 복합재난 피해 예측 및 신속한 대응이 가능하며 이러한 기술격차를 해소하기 위해서는 선진 연구기관의 공동연구 등을 통한 한국 환경에 최적화된 기술 개발이 필요함.



[재난 및 안전관리기술개발 종합계획]

- 해당 시스템의 경우 시설물 관련 데이터를 학계, 산업체 등 일반에 실시간 공유하여 다양한 복합 데이터, 빅데이터 연계 활용 기회를 제공하고 있음.
- 미국의 경우 자연재난 담당기구인 연방재난관리청(Federal Emergency Management Agency: FEMA) 운영, ICT 관련 기술의 경우 재난관리용 통신시스템 및 GIS 기반 재난관리시스템인 Hazus-MH를 이용해 지진·허리케인·홍수 등 3개 자연재해에 대응하는 구조물(건물, 전기, 교통, 댐, 핵발전소, 제방 등의 사회기반시설)에 대한 GIS 기반의 ‘위험도 맵’을 제공.
- 일본은 ‘일본방재백서(2010)’에서 과학기술연구, 국토보전, 재해복구 등 방재 관련 대책 및 예산을 수립 했으며 방재기술 대안으로 홍수, 내수, 고조, 해일, 토사재해, 화산, 지진 등 ‘해저드 맵’의 활용을 적극 추진.
- EU는 유럽 전체 재난위험도 평가 시스템인 ESPON(European Spatial Planning Observation Network)을, 대만에서는 HAZ-Taiwan 프로젝트 등을 통해 ICT 기반 재난관리체계를 구축해 운영하는 등 전세계적으로 재난관리에 대비.
- 선진국의 경우 Enterprise GIS, 통합 GIS 등 국가지리정보서비스 기반의 실시간 재난정보, 과거 재난기록 표시 등을 재난상황 파악에 활용함.
- 선진국들의 경우 데이터 활용을 위한 정보 및 운영 체계가 잘 정비되어 있지만 실시간 3차원 영상과 결합하여 연계 데이터를 관리하는 체계는 아직은 초보 수준임.

• 국외 표준화 현황

- 국제표준화기구 재난관리기술위원회(ISO/TC223)에서 국제재난관리 표준 제정을 위한 활동을 추진함.
 - * ISO22300(용어) : 스웨덴, 네덜란드 등 14개국 도입
 - * ISO22301(업무연속성관리시스템-요구) : 영국, 태국 등 16개국 도입
 - * ISO22320(재난관리-사고대응 요구) : 독일, 태국 등 15개국 도입
 - * ISO22351(정보공유를 위한 데이터 요소와 코드) : 프랑스
- 미국, 영국, 일본 등 선진국에서는 재난관리의 중요성을 인식하여 1990년대부터 재난관리

표준화를 추진함.

- * 미국은 미연방재난관리청, 미국재난관리자협회, 국제재난관리자협회가 협력하여 지속적으로 개정(NFPA 1600 : 재난관리 및 응급관리 체제와 사업연속성 확보 제체의 표준)하고 NFPA 1600을 국가표준으로 국제표준화를 추진함.

제3장. 연구수행내용 및 성과

1. SOC 시설물의 스마트 통합 유지 관리를 위한 프레임워크의 설계

시설물 관리 체계에 대한 자동화, 표준화, 효율성을 제고하기 위해서는 전체 소프트웨어를 포함하는 전체 프로세스의 소프트웨어 프레임워크를 정의해야함. 개발 시스템의 지속적인 운영 효율성 제고를 위해 프로세스의 개념을 적용하여 시설물 관리에 대한 프로세스의 구성요소인 드론 탑재 3D엔진과 무선 제어 측정 플랫폼을 포함하여 프레임워크를 설계함. 이를 통해 다양한 시설물 관리에 유연하게 적용하며, 통합 관리의 효율성을 높일 수 있음.

다음은 본 연구의 목표 시스템에 대해 시설물 관리 통합 프로세스 방법론에 대해 연구된 내용을 정리하였다.

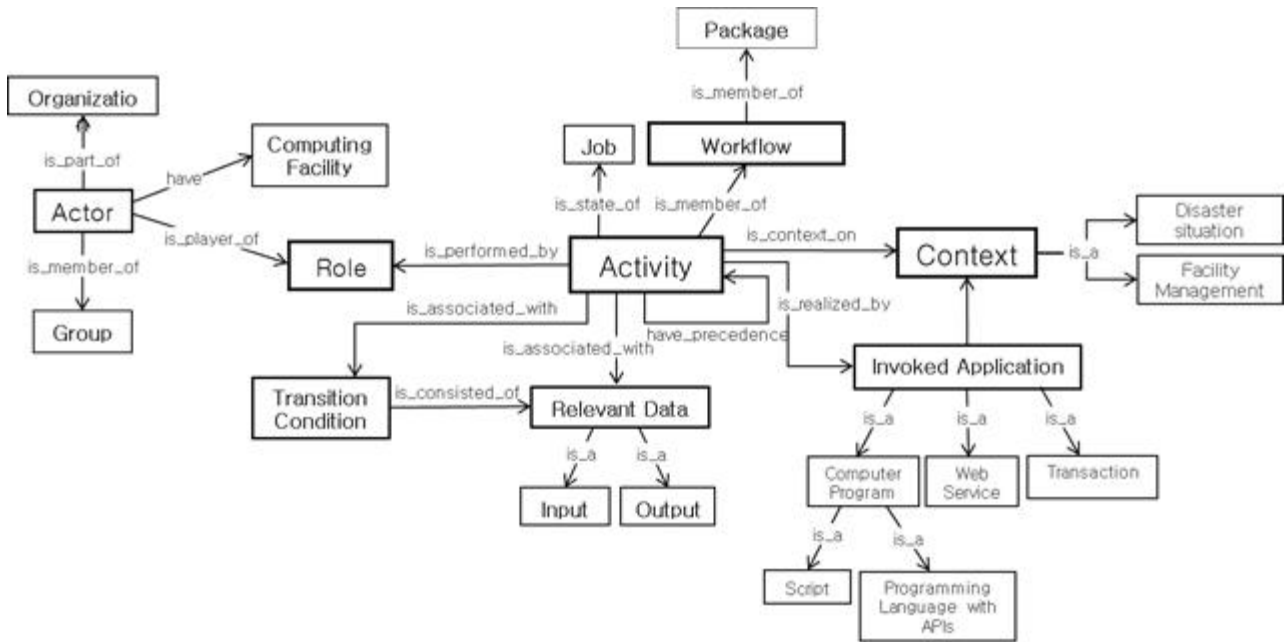
(1) 시설물 관리 프로세스 모델링 방법 연구 및 개발

시설물 관리 프로세스를 모델링하기 위해서는 시설물 관리를 위한 상황과 요소에 대한 정리를 위한 모델링 방법이 필요하다. 따라서 여기서는 시설물 관리 프로세스 모델을 위한 메타모델을 정의한다.

시설물 관리 프로세스 모델의 메타-모델은 시설물 관리 프로세스 모델을 생성하기 위한 방법으로써, 각 단위 구성의 속성을 정의할 수 있으며 다음은 시설물 관리를 위한 프로세스 모델의 주요 단위 요소들이다.

- 액티비티(Activity):
 - 시설물 관리를 위한 모든 구성요소들은 단위 프로세스로 묶여지게 된다.
 - 각 단위 프로세스의 액티비티는 그 특성에 따라, 시설물 관리 방법의 정의된 정보에 따라 수행 주체가 결정되게 됨.
 - 수행 주체는 일반적으로 시설물 관리를 위한 사람 및 조직(업무 수행 후보자 그룹)이 수행하게 될 수 있음.
 - 그리고 상황 정보 인식을 통해 인식 정보에 따라, 재난 상황과 일반상황으로 나누어 관리 될 수 있음.
 - 이 상황 정보에 대한 처리는 애플리케이션 및 서비스를 통해 자동화 하여 처리 가능함.

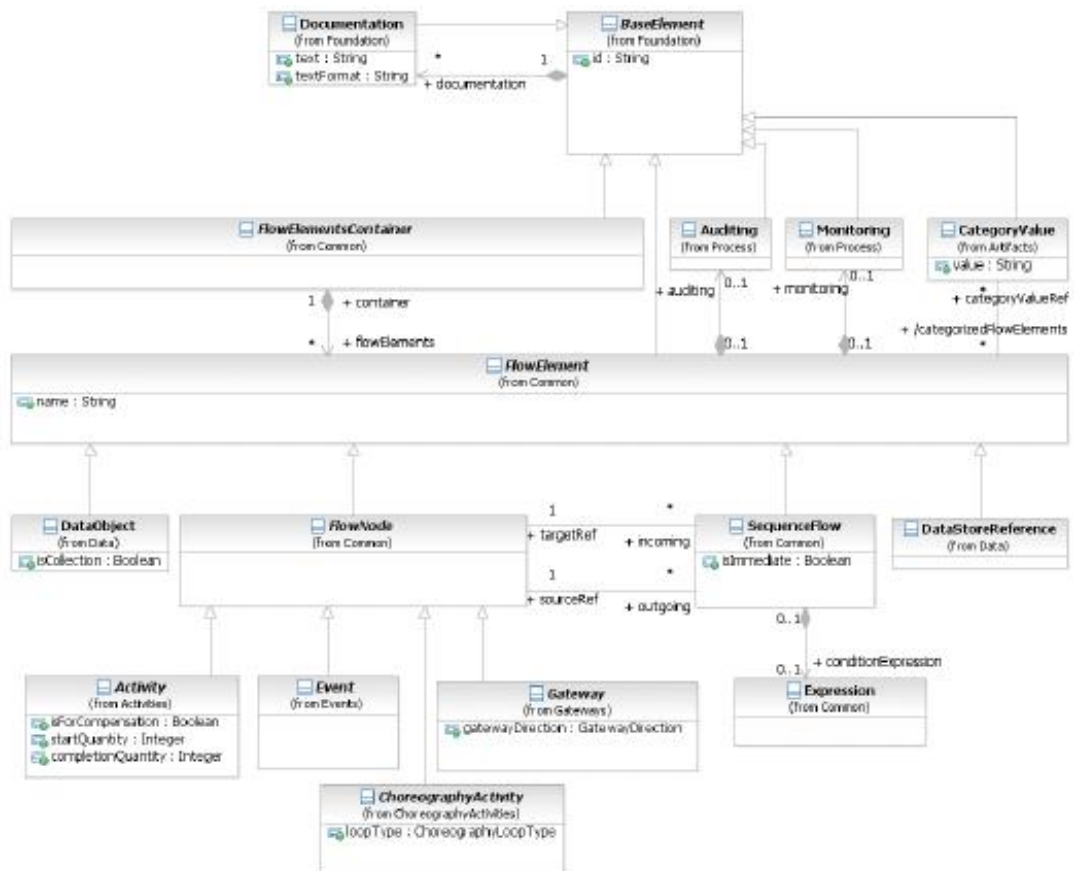
- 역할(Role):
 - 시설물 관리를 위한 관리 주체에 대한 정의를 하게 됨.
 - 관리 주체는 사람 및 조직이 될 수 있는데, 이때 정보의 입력과 처리를 위한 상세 정보 등을 관리 할 수 있도록 모델링 방법과 시스템에서 지원 가능해야함.
- 상황(Context):
 - 상황 정보는 크게 일반 관리 상황과 재난 상황으로 구분되어 관리함.
 - 일반 상황에서는 주기적으로 관리 될 수 있는 시설물 관리에 대한 관리 포인트를 정의하고 스케줄링을 통해 관리함.
 - 재난 상황에 대한 정의는 예측 가능한 부분과 예측 불가능한 부분에 모두 대응 할 수 있도록 상황 정보에 대한 기술을 제공하도록 함.
 - 상황 정보에 대해 처리해야할 부분은 자동화된 애플리케이션 및 서비스를 통해 처리 가능하도록 함.
- 제어(Control/Transition Condition):
 - 전반적인 상황의 흐름에 대한 액티비티의 흐름은 정해진 제어(이동 경로 및 조건)에 따라 정의 될 수 있음.
 - 상황에 대한 판단과 의사결정을 통해, 적절한 방향으로 시설물 관리를 할 수 있도록 정보를 제공함.
- 애플리케이션(Invoked Application):
 - 일반 상황에 대한 처리를 위한 규칙이나 방법에 대한 정의를 제공할 수 있음.
 - 재난 상황에 대한 인식과 처리를 다양한 인지 방법과 처리 및 서비스를 제공할 수 있음.



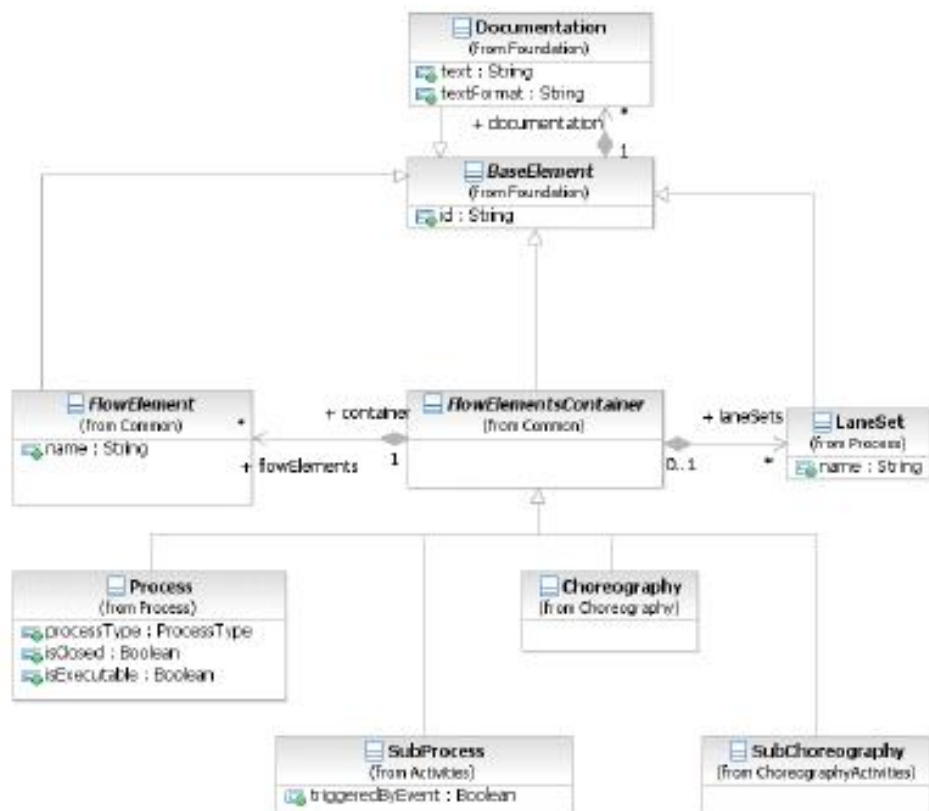
[시설물 관리 프로세스 모델의 메타-모델]

(2) 시설물 관리 프로세스 모델링 시스템 구현

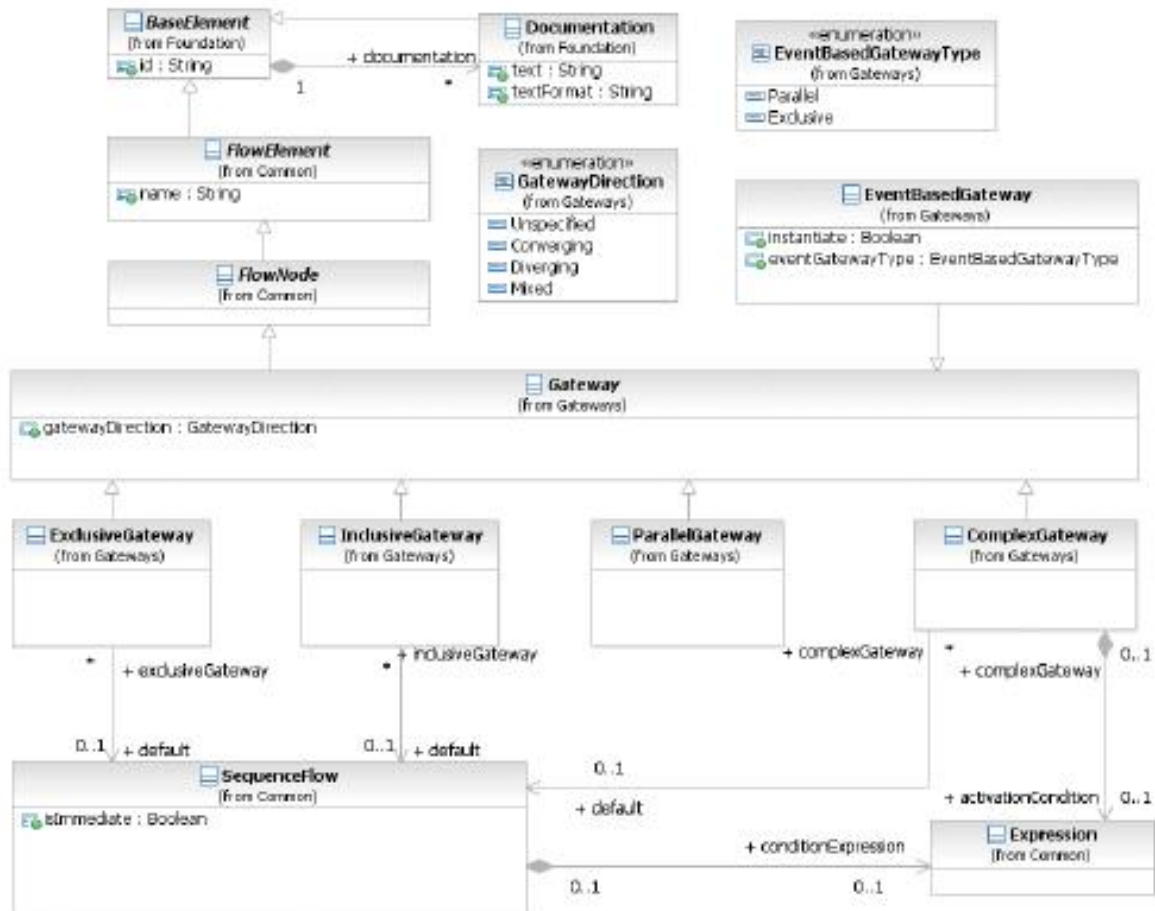
- 프로세스를 기반으로 한 시설물 관리에 대한 모델을 정의하고 실행시키기 위한 방법론을 기반으로 한 시설물 관리 프로세스 모델링 시스템을 구현함.
 - 시설물 관리 프로세스 모델링 시스템은 일반적인 프로세스 구성 요소 운영 방법과 새로운 형태의 프로세스 구성 요소 운영 방법을 복합하여 서비스를 제공할 수 있도록 함.
 - 따라서, 일반적인 프로세스 모델링 방법으로 잘 알려진 BPMN을 연구 확장 및 활용함.
- BPMN 2.0
 - BPMN은 프로세스를 정의하기 위한 다양한 단위 요소와 패턴을 가지고 있음.
 - 본 연구에서는 잘 알려진 BPMN의 2.0 버전을 가지고 적용하였음.
 - 일반적인 프로세스의 정의를 위해, 프로세스(플로우) 요소(속성)의 관계를 정의함.
 - 프로세스의 제어 흐름을 정리하기 위해 프로세스(플로우) 조건의 관계를 정의함.
 - 제어 흐름의 복잡도를 정리하기 위해 게이트웨이 구성의 관계를 정의함.



[프로세스(플로우) 요소의 관계]

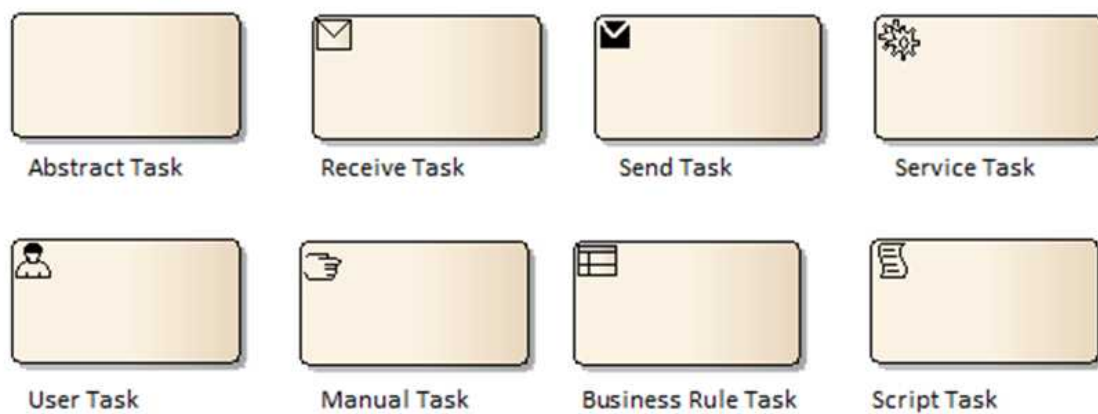


[프로세스(플로우) 조건의 관계]



[게이트웨이 구성의 관계]

- 프로세스 정의를 위해 업무(Task/액티비티) 구성을 정의함.



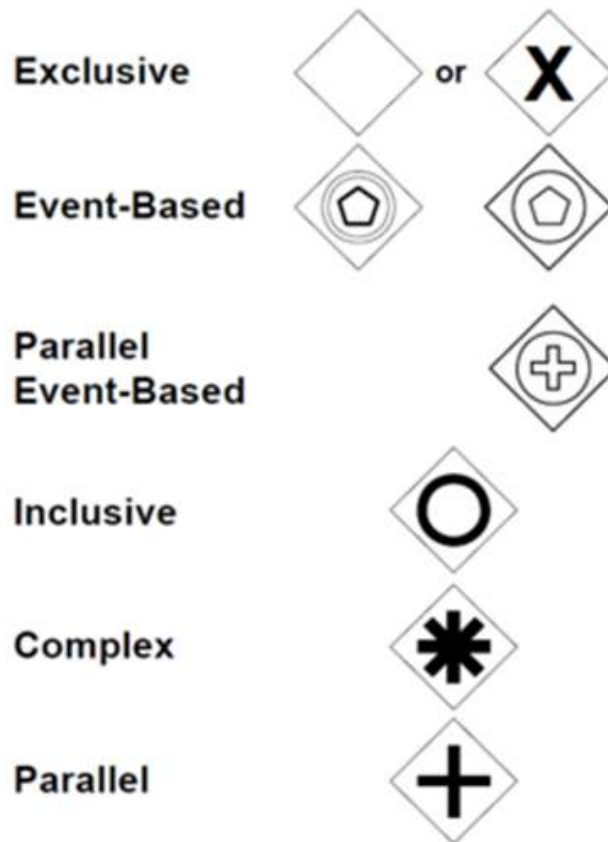
[업무(Task/액티비티) 구성]

- 프로세스 정의를 위해 이벤트 구성을 정의함.

	"Catching"		"Throwing"		Non-Interrupting	
Message						
Timer						
Error						
Escalation						
Cancel						
Compensation						
Conditional						
Link						
Signal						
Terminate						
Multiple						
Parallel Multiple						

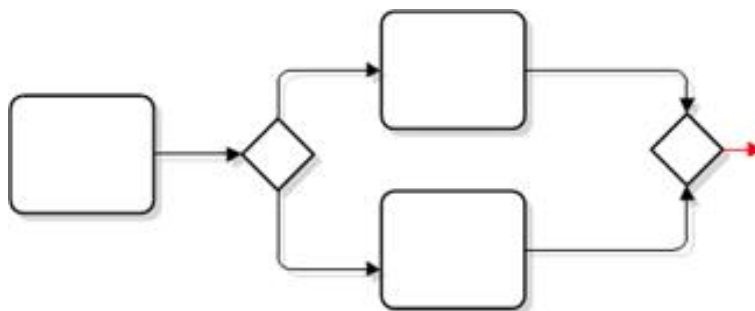
[이벤트 구성]

- 프로세스 정의를 위해 분기를 위한 게이트웨이 구성을 정의함.

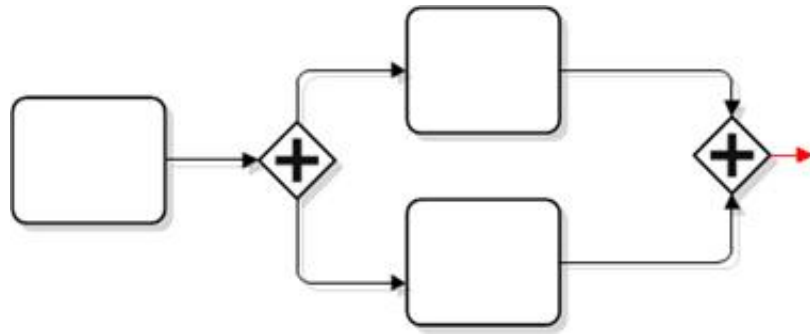


[게이트웨이 구성]

- 프로세스 정의의 배타적 분기와 병렬 분기 방법을 정의함.



[배타적 분기 방법]



[병렬 분기 방법]

- BPMN 2.0의 확장

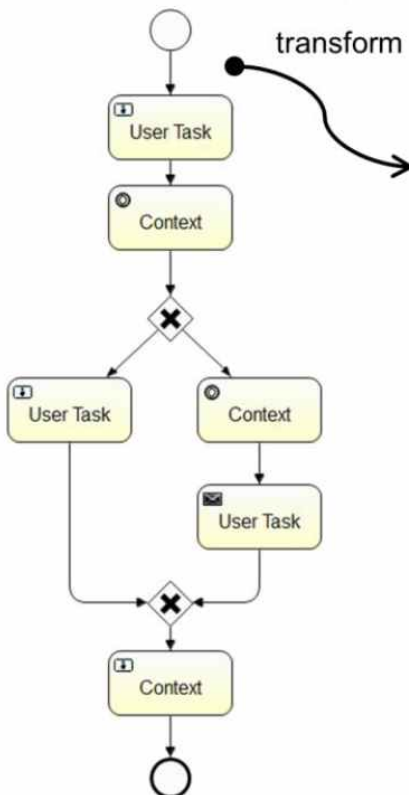
- BPMN 2.0의 단위 구성 요소인 업무(Task/액티비티)에 초점을 맞추도록 함.
- BPMN 2.0의 단위 요소는 서로 확장될 수 있으며, 확장에 대한 방법을 제공하도록 하고 있음.
- 기본 구성요소에 상황 정보에 대한인지를 할 수 있도록, 확장 노드를 추가함.
- 확장의 경우, 일반 상황과 재난 상황에 대한 구분과 처리를 위한 애플리케이션 및 서비스를 처리하기 위한 노드로의 확장을 정의함.
- 시설물 관리에 대한 고려 사항을 정의하고 처리하기 위해 상황 정보 인식 노드로 단위 구성요소를 추가함.
- 시설물 관리 프로세스를 정의하기 위한 모델링 도구 및 시스템을 구현하기 위해, 기본 노드와 확장 노드를 함께 사용 가능하도록 정의 .



[BPMN 2.0의 확장 - 프로세스 기반 시설물 관리를 위한 상황 정보 인식 노드의 확장]

- 확장된 BPMN 2.0 모델링 시스템

- 확장된 BPMN 2.0 방법에 따라 시설물 관리 프로세스 모델을 구현하기 위해 모델링 시스템을 확장 구현함.
- 확장된 BPMN 2.0 모델링 시스템은 확장 노드의 표현을 지원함.
- 확장 노드의 표현을 통해, 시설물 관리를 위한 상황 정보에 대해 정의할 수 있음.
- 정의된 시설물 상황에 대한 정보는 추후, 운영 시스템을 통해 운영 될 수 있음.
- 시스템이 인지하고 운영하기 위해, BPMN 2.0 기반의 확장 프로세스 모델은 XML 기반의 프로세스 실행 정보를 변환 되어 처리함.
- 변환된 XML 기반의 프로세스 정보는 프로세스 해석 도구를 통해, 해석 가능함.
- 프로세스 정보의 해석을 통해, 자동화된 프로세스의 운영과 관리의 자동화를 실현할 수 있음.
- 시설물 관리를 위한 사용자의 행위와 상황에 대한 인식을 기반으로 정해진 프로세스의 방향에 따라 수행.



```

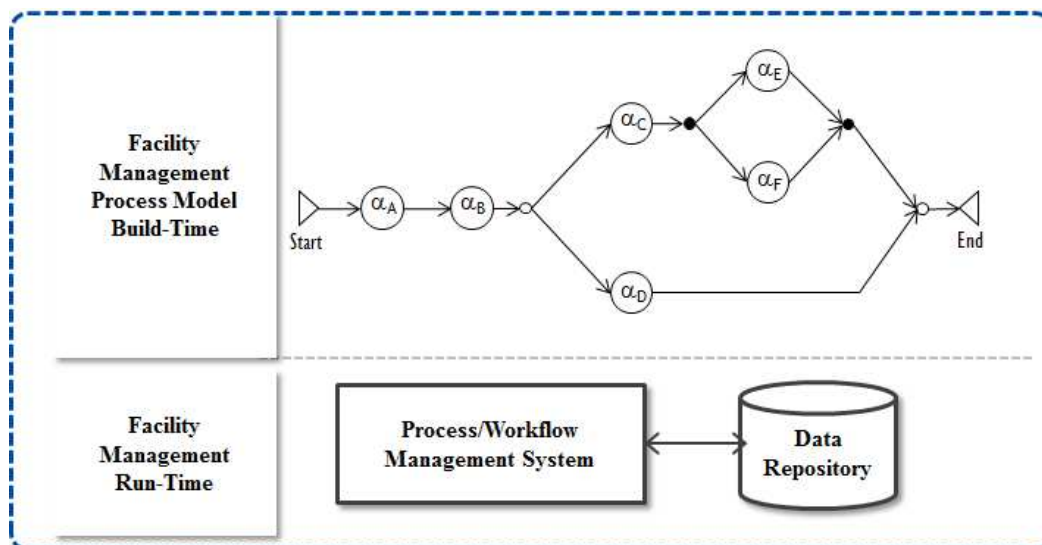
1 <?xml version="1.0" encoding="UTF-8"?>
2 <definitions
3   xmlns="http://www.omg.org/spec/BPMN/20100524/MODEL"
4   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5   xmlns:xs="http://www.w3.org/2001/XMLSchema"
6   xmlns:activiti="http://activiti.org/bpmn"
7   xmlns:bpmndi="http://www.omg.org/spec/BPMN/20100524/DI"
8   xmlns:omgdc="http://www.omg.org/spec/DC/20100524/DC"
9   xmlns:omgdi="http://www.omg.org/spec/DC/20100524/DI" typeLanguage="http://www.w3.org/2001/XMLSchema" expressionLanguage="http://www.w3.org/2001/XMLSchema"
10 >
11 <process id="myProcess" name="My process" isExecutable="true">
12   <startEvent id="startevent1" name="Start"></startEvent>
13   <userTask id="usertask1" name="User Task"></userTask>
14   <sequenceFlow id="flow1" sourceRef="startevent1" targetRef="usertask1"></sequenceFlow>
15   <serviceTask id="servicetask1" name="Context" activiti:extensionId="ctrl.ex_icn.servicetasks.ioIActivity"></serviceTask>
16   <sequenceFlow id="flow2" sourceRef="usertask1" targetRef="servicetask1"></sequenceFlow>
17   <exclusiveGateway id="exclusivegateway1" name="Exclusive Gateway"></exclusiveGateway>
18   <sequenceFlow id="flow3" sourceRef="servicetask1" targetRef="exclusivegateway1"></sequenceFlow>
19   <userTask id="usertask2" name="User Task"></userTask>
20   <sequenceFlow id="flow4" sourceRef="exclusivegateway1" targetRef="usertask2"></sequenceFlow>
21   <serviceTask id="servicetask2" name="Context" activiti:extensionId="ctrl.ex_icn.servicetasks.ioIActivity"></serviceTask>
22   <sequenceFlow id="flow5" sourceRef="usertask2" targetRef="servicetask2"></sequenceFlow>
23   <sequenceFlow id="flow6" sourceRef="servicetask2" targetRef="mailtask1"></sequenceFlow>
24   <exclusiveGateway id="exclusivegateway2" name="Exclusive Gateway"></exclusiveGateway>
25   <sequenceFlow id="flow7" sourceRef="usertask2" targetRef="exclusivegateway2"></sequenceFlow>
26   <sequenceFlow id="flow8" sourceRef="mailtask1" targetRef="exclusivegateway2"></sequenceFlow>
27   <userTask id="usertask3" name="Context"></userTask>
28   <sequenceFlow id="flow9" sourceRef="exclusivegateway2" targetRef="usertask3"></sequenceFlow>
29   <endEvent id="endevent1" name="End"></endEvent>
30   <sequenceFlow id="flow10" sourceRef="usertask3" targetRef="endevent1"></sequenceFlow>
31 </process>
32 <bpmndi:BPMNDiagram id="BPMNDiagram_myProcess">
33   <bpmndi:BPMNPlane bpmnElement="myProcess" id="BPMNPlane_myProcess">
34     <bpmndi:BPMNShape bpmnElement="startevent1" id="BPMNShape_startevent1">
35       <omgdc:Bounds height="35.0" width="35.0" x="220.0" y="40.0"></omgdc:Bounds>
36     </bpmndi:BPMNShape>
37     <bpmndi:BPMNShape bpmnElement="usertask1" id="BPMNShape_usertask1">
38       <omgdc:Bounds height="55.0" width="105.0" x="185.0" y="112.0"></omgdc:Bounds>
39     </bpmndi:BPMNShape>
40     <bpmndi:BPMNShape bpmnElement="servicetask1" id="BPMNShape_servicetask1">
41       <omgdc:Bounds height="55.0" width="105.0" x="185.0" y="188.0"></omgdc:Bounds>
42     </bpmndi:BPMNShape>
43     <bpmndi:BPMNShape bpmnElement="exclusivegateway1" id="BPMNShape_exclusivegateway1">
44       <omgdc:Bounds height="40.0" width="40.0" x="217.0" y="260.0"></omgdc:Bounds>
45     </bpmndi:BPMNShape>
46     <bpmndi:BPMNShape bpmnElement="usertask2" id="BPMNShape_usertask2">
47       <omgdc:Bounds height="55.0" width="105.0" x="180.0" y="350.0"></omgdc:Bounds>
48     </bpmndi:BPMNShape>
49     <bpmndi:BPMNShape bpmnElement="servicetask2" id="BPMNShape_servicetask2">
50       <omgdc:Bounds height="55.0" width="105.0" x="260.0" y="350.0"></omgdc:Bounds>
51     </bpmndi:BPMNShape>
52     <bpmndi:BPMNShape bpmnElement="mailtask1" id="BPMNShape_mailtask1">
53       <omgdc:Bounds height="55.0" width="105.0" x="260.0" y="440.0"></omgdc:Bounds>
54     </bpmndi:BPMNShape>
55     <bpmndi:BPMNShape bpmnElement="exclusivegateway2" id="BPMNShape_exclusivegateway2">

```

[확장된 BPMN 2.0기반 시설물 관리 프로세스 모델의 예]

(3) 시설물 관리 프로세스 운영 시스템 구현

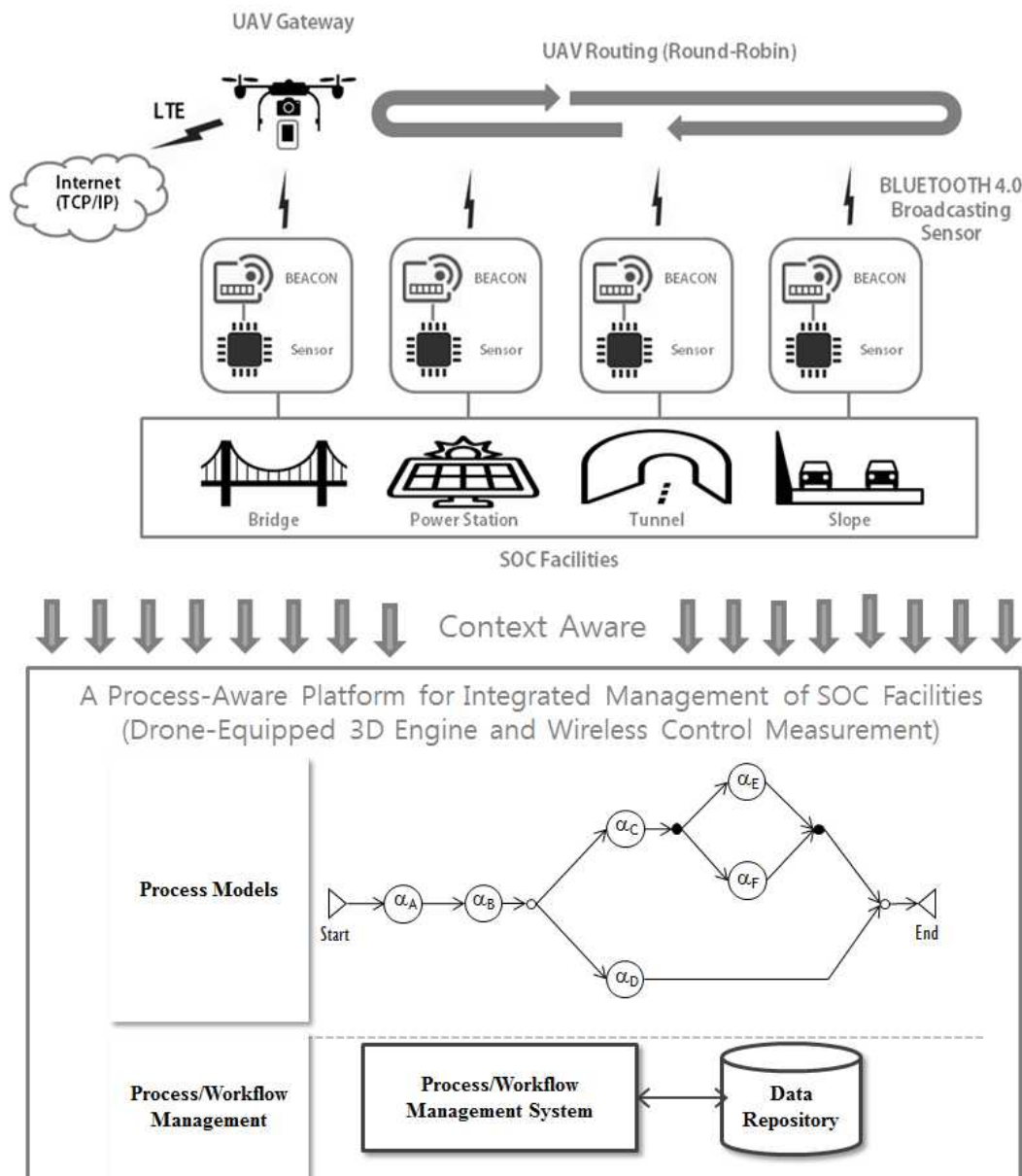
- 시설물 관리 프로세스의 운용은 크게 두 가지 관점으로 분리될 수 있음.
- 첫 번째 관점은 시설물 관리 프로세스 모델의 빌드타임으로 프로세스의 정의와 관련이 있음.
- 두 번째 관점은 시설물 관리의 런타임으로 프로세스의 실행과 관련이 있음.
- 시설물 관리 프로세스 운영 시스템은 프로세스 관리 시스템의 운영과 그에 대한 데이터 관리로 볼 수 있음.
- 데이터는 모델 정보에 대한 실행 분기 정보를 관리하는 부분과 런타임 워크플로우 관련 데이터 부분으로 구분 될 수 있음.



[시설물 관리 프로세스 운영 - 빌드 시점과 실행 시점]

(4) 시설물 통합 관리를 위한 프로세스 기반 플랫폼

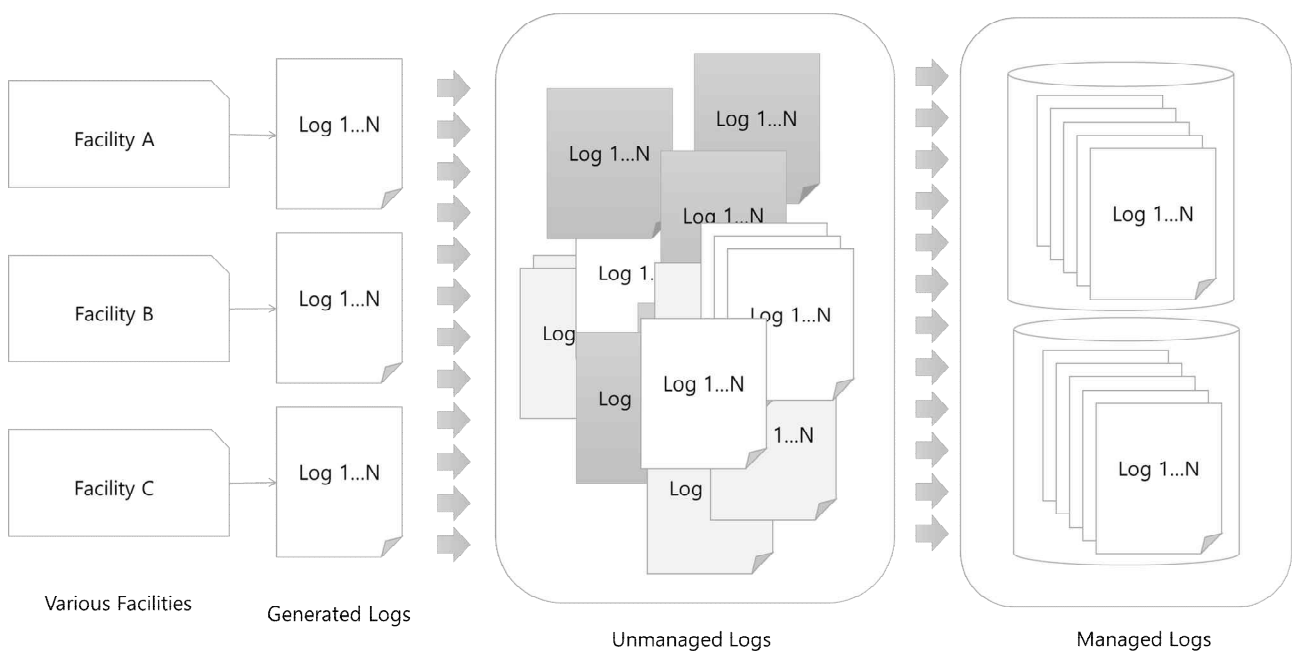
- 시설물 관리 프로세스 운영에 대해, 시설물에 대한 전반적인 데이터 상황 인식에 대응하도록 구성함.
- 시설물 관리를 위해 UAV 게이트웨이와 라우팅 기술 등을 통해 관련 통신 인프라를 구성함.
- 구성된 통신 인프라는 시설물 지원을 위한 센서와 연동하여 구성됨.
- 즉, 시설물과 관련된 상황정보와 인식과 데이터 처리 방법을 구성하게 됨.
- 그 상황정보 인식 및 데이터 처리 방법에 대해, 프로세스 모델과 실행 구성요소를 접목한 통합 플랫폼을 구성함.



[시설물 통합 관리를 위한 프로세스 기반 플랫폼]

(5) 시설물 관리 프로세스 운영 파일 관리 도구의 구현

- 시설물 데이터는 각 시설물의 종류나 형태에 따라, 서로 다른 형태와 종류의 데이터를 만들어내게 됨.
- 서로 다른 형태와 종류의 데이터는 통합된 프로세스 관리 플랫폼을 통해, 운영되기 위해, 실행 기록들을 관리해야함.
- 이러한 서로 다른 기록의 데이터를 관리된 실행 정보로 묶기 위한 통합 데이터 관리 방법이 필요함.
- 이러한 통합 데이터의 관리를 통해, 추후 시설물 관리 프로세스의 개선 시 유용하게 사용될 수 있음.



[시설물 데이터 관리를 위한 데이터 통합 관리 방법]

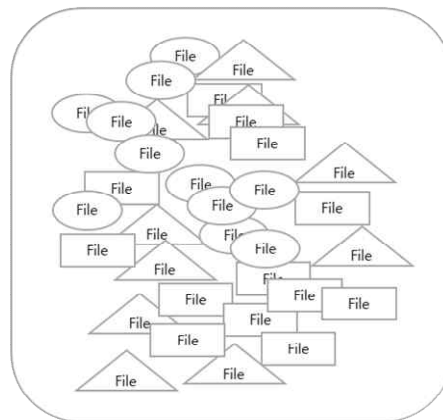
- 이때, 다양한 종류와 형태의 시설물들의 데이터를 처리하기 위한 규칙 정의함.
- 정의된 규칙에 따라, 다양하고 복잡한 데이터는 정리될 수 있음.
- 이렇게 정리된 데이터를 활용하여, 추후 프로세스 개선에 활용함.

Conditions	Rules
Condition 1	Include
Condition 2	Do not include
Condition 3	Equal
Condition 4	Do not equal
Condition 5	Start together
Condition 6	Do not end with

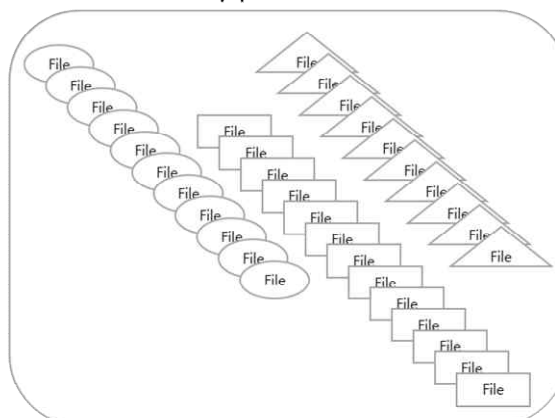
[시설물 데이터 관리를 위한 규칙의 예]

- 정해진 규칙에 따라, 시설물 관리 시 발생하는 기록 데이터에 대한 파일은 관리 될 수 있음.

Unmanaged Files of Various Facilities

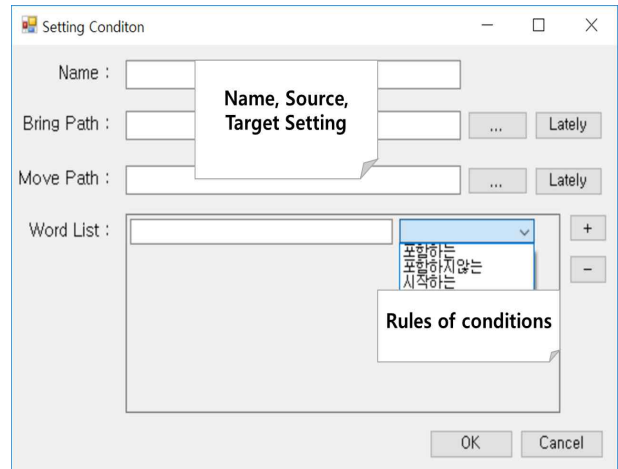
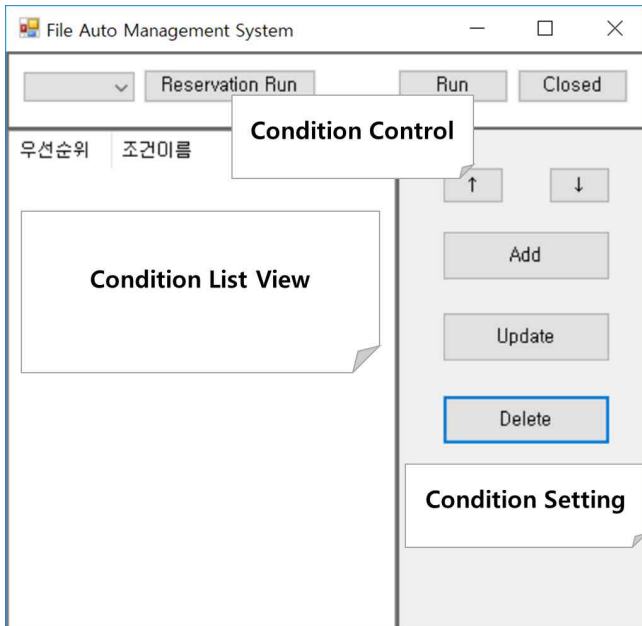


Files applied Rules



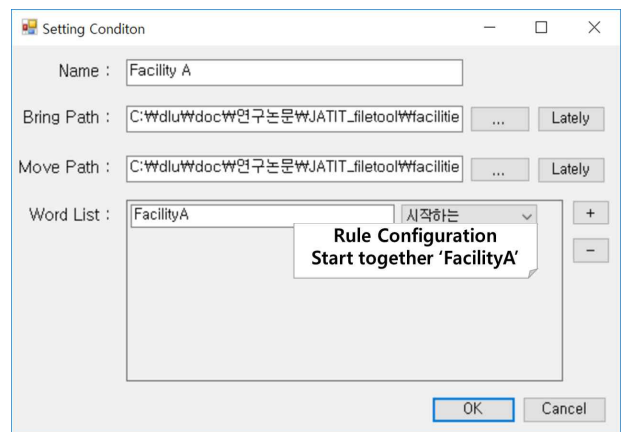
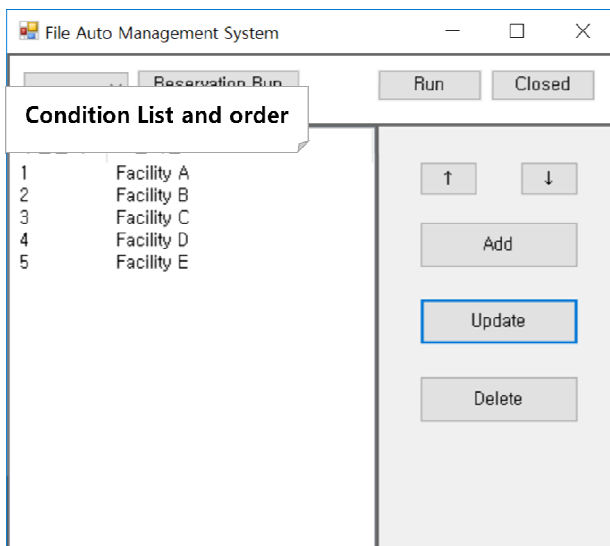
[규칙 기반 데이터 통합]

- 정해진 규칙에 따라, 시설물 관리 시 발생하는 기록 데이터에 대한 파일은 관리 될 수 있음.
- 이러한 규칙은 관리 도구를 통해 구현될 수 있음.



[규칙 기반 데이터 통합 도구]

- 규칙정의 및 실행은 도구를 통해 처리되며, 이러한 규칙에 따라, 시설물 관리 정보에 대한 기록은 통합 관리 될 수 있음.



[규칙 기반 데이터 통합 도구 수행의 예 - 정의]

```

C:\WINDOWS\system32\cmd.exe
C:\Wdlu\doc\연구논문\JATIT_filetool\facilities\managed>dir
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: A080-0879

C:\Wdlu\doc\연구논문\JATIT_filetool\facilities\managed 디렉터리
2018-02-04 오전 03:49 <DIR> .
2018-02-04 오전 03:49 <DIR> ..
2018-02-04 오후 12:31 <DIR> FacilityA
2018-02-04 오후 12:31 <DIR> FacilityB
2018-02-04 오후 12:31 <DIR> FacilityC
2018-02-04 오후 12:31 <DIR> FacilityD
2018-02-04 오후 12:31 <DIR> FacilityE
0개 파일
0 바이트
7개 디렉터리 64,286,085,120 바이트 남음

C:\Wdlu\doc\연구논문\JATIT_filetool\facilities\managed>

C:\Wdlu\doc\연구논문\JATIT_filetool\facilities\managed\FacilityA 디렉터리
2018-02-04 오후 12:31 <DIR> .
2018-02-04 오후 12:31 <DIR> ..
2018-02-03 오후 02:54 0 FacilityA001.txt
2018-02-03 오후 02:54 0 FacilityA002.txt
2018-02-03 오후 02:54 0 FacilityA003.txt
2018-02-03 오후 02:54 0 FacilityA004.txt
2018-02-03 오후 02:54 0 FacilityA005.txt
2018-02-03 오후 02:54 0 FacilityA006.txt
0개 파일
0 바이트
2개 디렉터리 64,286,085,120 바이트 남음

C:\Wdlu\doc\연구논문\JATIT_filetool\facilities\managed\FacilityA>

C:\Wdlu\doc\연구논문\JATIT_filetool\facilities\managed\FacilityB 디렉터리
2018-02-04 오후 12:31 <DIR> .
2018-02-04 오후 12:31 <DIR> ..
2018-02-03 오후 02:54 0 FacilityB001.txt
2018-02-03 오후 02:54 0 FacilityB002.txt
2018-02-03 오후 02:54 0 FacilityB003.txt
2018-02-03 오후 02:54 0 FacilityB004.txt
2018-02-03 오후 02:54 0 FacilityB005.txt
2018-02-03 오후 02:54 0 FacilityB006.txt
0개 파일
0 바이트
2개 디렉터리 64,286,085,120 바이트 남음

C:\Wdlu\doc\연구논문\JATIT_filetool\facilities\managed>

C:\Wdlu\doc\연구논문\JATIT_filetool\facilities\managed\FacilityC 디렉터리
2018-02-04 오후 12:31 <DIR> .
2018-02-04 오후 12:31 <DIR> ..
2018-02-03 오후 02:54 0 FacilityC001.txt
2018-02-03 오후 02:54 0 FacilityC002.txt
2018-02-03 오후 02:54 0 FacilityC003.txt
2018-02-03 오후 02:54 0 FacilityC004.txt
0개 파일
0 바이트
2개 디렉터리 64,286,089,216 바이트 남음

C:\Wdlu\doc\연구논문\JATIT_filetool\facilities\managed>

C:\Wdlu\doc\연구논문\JATIT_filetool\facilities\managed\FacilityD 디렉터리
2018-02-04 오후 12:31 <DIR> .
2018-02-04 오후 12:31 <DIR> ..
2018-02-03 오후 02:54 0 FacilityD001.txt
2018-02-03 오후 02:54 0 FacilityD002.txt
2018-02-03 오후 02:54 0 FacilityD003.txt
2018-02-03 오후 02:54 0 FacilityD004.txt
0개 파일
0 바이트
2개 디렉터리 64,286,089,216 바이트 남음

C:\Wdlu\doc\연구논문\JATIT_filetool\facilities\managed>

C:\Wdlu\doc\연구논문\JATIT_filetool\facilities\managed\FacilityE 디렉터리
2018-02-04 오후 12:31 <DIR> .
2018-02-04 오후 12:31 <DIR> ..
2018-02-03 오후 02:54 0 FacilityE002.txt
2018-02-03 오후 02:54 0 FacilityE004.txt
0개 파일
0 바이트
2개 디렉터리 64,286,089,216 바이트 남음

C:\Wdlu\doc\연구논문\JATIT_filetool\facilities\managed>

```

[규칙 기반 데이터 통합 도구 수행의 예 - 실행]

- 이렇게 통합 관리 될 수 있는 시설물 관리 실행 이력 정보는 프로세스 개선을 수립할 수 있음.
- 개선된 프로세스를 통해, 시설물 통합관리 프로세스 플랫폼에서 운용될 수 있음.

2. 현장용 무선통신 기반 계측센서 다중 인터페이스 연동 모듈 개발

(1) 블루투스 비콘 계측기 분석 및 적용

○ Midascon 비콘 연동 모듈 개발

- Midascon 비콘을 이용하여 주변 환경의 데이터를 수신 받아 다시 서버로 전송하는 스마트폰 용 앱을 개발 하였다. Midascon 비콘은 산업 안전관리용 장비로 온도, 습도 등을 측정할 수 있는 센서들이 내장되어있어 스마트폰과 블루투스를 지원하는 장치와 통신하여 그 계측치를 전송한다.



○ Midascon 비콘 계측기 사양

- Midascon 비콘의 블루투스 기능은 기기의 배터리 수명에 영향을 거의 미치지 않고 설정 값에 따라 최대 2년까지 사용할 수 있는 저전력 기기이며 Midascon은 BLE 기술을 이용한 신호를 송출하는 기기로서, 코인 배터리를 사용하여 휴대 및 부착이 용이하다. 또한 Midascon은 다양한 센서(온도, 습도, 가속도)도 함께 내장하고 있어, 재난안전 및 시설물 유지 관리에 활용이 가능하다.

Item	Details
Accelerometer Sensor	Measurement range : FS bit set to 00 (± 2.0)
Certification	KC 인증
Button	Emergency alarm
Battery	CR2450 Coin Battery (3v/620mAh); Up to 2 years
RTC[Real Time Clock]	Built-in clock correction function
Wireless Technology	Bluetooth® Smart 4.1(CSR1010 chipset)
Temperature & Humidity Sensor	Accuracy : $-10^{\circ}\text{C} < t_A < 85^{\circ}\text{C}$ (Typ ± 0.3) 0 – 80% RH(Typ ± 2)
Size / Weight / Color	32mm x 66mm x 7.9mm / 18g / Ivory
Storage temperature range	$-40^{\circ}\text{C} \sim 85^{\circ}\text{C}$
Operating temperature range	$-20^{\circ}\text{C} \sim 60^{\circ}\text{C}$

[Midascon 비콘 사양]

○ 비콘 계측기 개발용 SDK

- Midascon 비콘 계측기는 안드로이드용 SDK를 제공하여 모바일 앱에 적용할 수 있다. 개발 언어는 자바를 사용하며 블루투스는 4.0을 지원한다.
- BLE(Bluetooth Low Energy)를 지원하는 안드로이드 4.3 , JAVA 1.7 이상부터 동작하며 권장사양은 안드로이드 5.0 이상 JAVA 1.7이다.

항목	내용
Android Target SDK Version	Android API Level 18 이상
Java Version	JDK 1.7
Sensor Module	Bluetooth 4.0 Le

[SDK 개발 환경]

○ 비콘 SDK API 제공 기능

• 비콘 API 제공 패키지

- com.hanvitsi.midascon : 비콘의 일반적인 기본 기능을 제공 하며 2개의 인터페이스와 5개의 클래스로 구성
- com.hanvitsi.midascon.manager : 비콘의 설정과 각종 관리 기능을 제공하며 4개의 클래스로 구성

- 패키지 com.hanvitsi.midascon

- 클래스 Beacon

메서드	설명
int	describeContents()
static double	distanceFromRssi(int rssi, int txPowerAtSource) 거리계산, 1미터 거리에서 측정 txPower 값을 조정하면서 측정한다.
int	getBattery() 배터리
android.bluetooth.BluetoothDevice	getDevice() 블루투스 디바이스
int	getEmergency() 응급 플래그
long	getEnterTime() 수신 종료 시간
long	getExitTime() 수신 시작 시간
float	getHumidity() 습도
java.lang.String	getId() ID
java.lang.String	getMac() Mac Address
int	getMajor() Major
java.lang.String	getMidasId() 마이더스 ID
int	getMinor() Minor
long	getRecordedTime() 업데이트 갱신 시간
int	getRssi() 신호세기
byte[]	getScanRecord() BLE 스캔 레코드
int	getStatus() 현재 상태
float	getTemperature() 온도
int	getType() 비콘 구분
long	getUpdateTime() 갱신 시간
java.lang.String	getUUID() UUID
void	updateScanRecord(int rssi, byte[] scanRecord) 비콘 업데이트
void	updateStatus(int status) 상태 업데이트
void	writeToParcel(android.os.Parcel dest,int flags)

- 클래스 BootCompletedReceiver

메서드	설명
void	onReceive(android.content.Context context, android.content.Intent intent)
	MidasService서비스실행

- 클래스 DebugConfig

메서드	설명
static void	d(java.lang.String message)
static void	e(java.lang.String message)
static void	i(java.lang.String message)
static void	v(java.lang.String message)
static void	w(java.lang.String message)

- 클래스 MidasApplication

메서드	설명
ContextManager	getContextManager()
android.content.SharedPreferences	getSharedPreferences(java.lang.String name,int mode)
void	onCreate()
void	setContextManager(ContextManager contextManager)

- 클래스 MidasService

메서드	설명
MidasApplication	getMidasApplication()
	MidasApplication호출
android.os.IBinder	onBind(android.content.Intent intent)
void	onCreate()
	ContextManager초기화
int	onStartCommand(android.content.Intent intent, int flags, int startId)
	서비스 주기 체크

- 패키지 com.hanvitsi.midascon.manager

- 클래스 BeaconSettings

메서드	설명
void	clearAllowBeacon()
	수신 비콘 설정 지우기
java.util.ArrayList<java.lang.String>	getAllowBeaconIdList()
	스캔 허용 비콘Id 리스트
int	getBeaconExitTime()
	비콘 이탈 시간
int	getLogWriteMode()
	로그 기록 모드
int	getServiceCheckDelay()
	서비스 체크 주기
boolean	isAllowBeacon(java.lang.String id)
	수신 비콘 확인
boolean	isAllowBeaconMac()
	Mac Address 비교 설정
boolean	isMidasScanMode()
	마이스다콘 스캔 모드
void	setAllowBeaconId(java.lang.String id, boolean insert)
	수신 비콘 설정
void	setAllowBeaconMac(boolean mac)
	Mac Address 비교 설정
void	setBeaconExitTime(int beaconExitTime)
	비콘 이탈 시간 설정
void	setLogWriteMode(int logWriteMode)
	로그 기록 모드
void	setMidasScanMode(boolean midasScanMode)
	마이스다콘 스캔 모드 설정
void	setServiceCheckDelay(int serviceCheckDelay, boolean update)
	서비스 체크 주기 설정

- 클래스 BeaconUtils

메서드	설명
static java.lang.String	createBeaconId(java.lang.String uuid, int major, int minor)
	비콘 ID 생성
static int[]	getAccelerometer(Beacon beacon)
	마이다스콘 가속도 센서 데이터 확인
static int	getBattery(byte[] scanRecord, int index)
	배터리 상태
static java.lang.String	getBeaconId(byte[] scanRecord)
	비콘 ID
static java.lang.String	getBeaconId(java.lang.String uuid, int major, int minor)
	비콘 ID
static int	getBeaconMajor(byte[] scanRecord)
	Major
static int	getBeaconMinor(byte[] scanRecord)
	Minor
static java.lang.String	getBeaconUUID(byte[] scanRecord)
	비콘 UUID
static int	getEmergency(byte[] scanRecord)
	응급신호
static float	getHumidity(byte[] scanRecord)
	습도
static java.lang.String	getMidasId(byte[] scanRecord)
	마이다스콘 ID
static java.lang.String	getMidasUUID(byte[] scanRecord)
	마이다스콘 UUID
static float	getTemperature(byte[] scanRecord)
	온도
static int	getTxPower(byte[] scanRecord, int index)
	TxPower
static boolean	isBeacon(byte[] scanRecord)
	비콘 여부 판단
static boolean	isMidascon(byte[] scanRecord)
	마이다스콘 판단

- 클래스 LogManager

메서드	설명
void	deleteBeaconLog()
	로그 삭제
static LogManager	getInstance(android.content.Context context)
void	onCreate(android.database.sqlite.SQLiteDatabase db)
void	onUpgrade(android.database.sqlite.SQLiteDatabase db,int oldVersion,int newVersion)
java.lang.String	selectBeaconLogAll()
	로그 조회

- 클래스 ContextManager

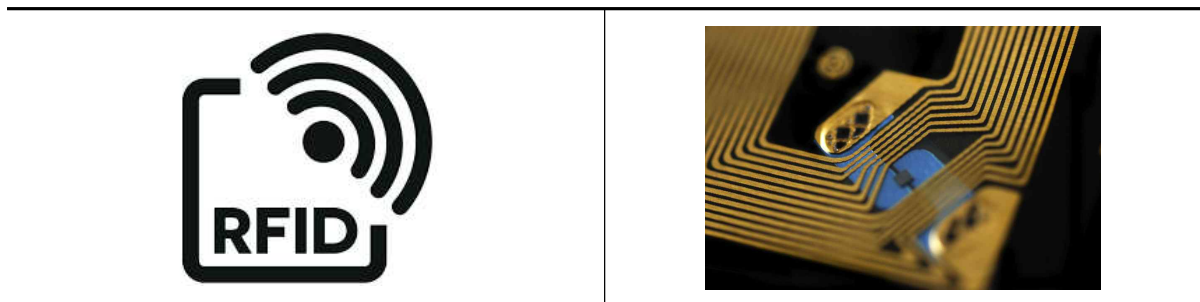
메서드	설명
static java.lang.String	createBeaconId(java.lang.String uuid, int major, int minor)
	비콘 ID 생성
static int[]	getAccelerometer(Beacon beacon)
	마이다스콘 가속도 센서 데이터 확인
static int	getBattery(byte[] scanRecord, int index)
	배터리 상태
static java.lang.String	getBeaconId(byte[] scanRecord)
	비콘 ID
static java.lang.String	getBeaconId(java.lang.String uuid, int major, int minor)
	비콘 ID
static int	getBeaconMajor(byte[] scanRecord)
	Major
static int	getBeaconMinor(byte[] scanRecord)
	Minor
static java.lang.String	getBeaconUUID(byte[] scanRecord)
	비콘 UUID
static int	getEmergency(byte[] scanRecord)
	응급신호
static float	getHumidity(byte[] scanRecord)
	습도
static java.lang.String	getMidasId(byte[] scanRecord)
	마이다스콘 ID
static java.lang.String	getMidasUUID(byte[] scanRecord)
	마이다스콘 UUID
static float	getTemperature(byte[] scanRecord)
	온도
static int	getTxPower(byte[] scanRecord, int index)
	TxPower
static boolean	isBeacon(byte[] scanRecord)
	비콘 여부 판단
static boolean	isMidascon(byte[] scanRecord)
	마이다스콘 판단

(2) 계측기와 연결되는 기기와의 연결 인터페이스 구축

○ 인터페이스 연결을 위한 통신 프로토콜 분석

- RFID(Radio-Frequency Identification)

- 주파수를 이용해 ID를 식별하는 SYSTEM으로 전자태그로 불린다. RFID는 전파를 이용해 정보를 수신하는 기술임.
- 근거리 통신망을 활용한 정보 취득 방식으로 주로 판매점의 제품 가격 등의 정보를 저장하고 있다. 리더기에 근접하면 정보를 읽을 수 있는 기기로 시설물 관리에서는 자재의 정보 등 기초 정보를 기록하고 제공하는데 활용됨.



[RFID 개념]

- 건축물의 유지관리 적용 분야로 통합관리시스템의 유지관리자재 입·출 실시간 관리에 활용되고 있으며, QR코드의 오염이 우려되는 부위 등 보완적인 역할 가능.
- NFC(Near Field Communication)
 - 근거리 무선 통신을 의미하며 아주 가까운 거리의 무선 통신을 위한 기술임.
 - 현재 데이터 통신 속도는 초당 424킬로비트 정도로 교통, 티켓, 지불 등의 분야에서 활용됨.
- Zigbee
 - Zigbee는 소형, 저전력 라디오를 이용해 개인 통신망을 구성하여 통신하는 표준기술을 의미함.

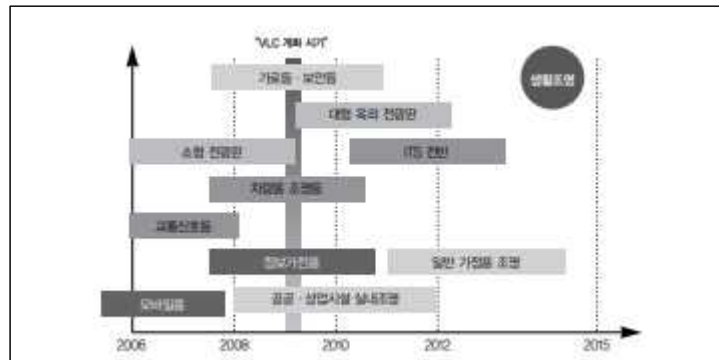


[ZigBee 통신 개념]

- ZigBee는 낮은 전력으로 구현 가능하며 자율적 무선 센서를 구축하여 무선 조명 스위치, 원격 검침 전력량계, 교통 관리 등에 활용됨.

- VLC(가시광 무선통신)

- 가시광 무선통신 기술은 백열전구와 형광등과 같은 조명이 디지털 반도체에 의한 LED(Light Emitting Diode) 조명으로 교체되는 인프라를 이용하여 통신을 가능하게 하는 기술로써 가시광 무선통신 PHY 기술, 가시광 무선통신 L2 MAC기술, 가시광 무선통신 응용 프로토콜 기술 등을 대상으로 하는 기술임.



- 블루투스(Bluetooth)

- 블루투스(Bluetooth)는 휴대폰, 노트북, 이어폰·헤드폰 등의 휴대기기를 서로 연결해 정보를 교환하는 근거리 무선 기술 표준을 뜻한다. 주로 10미터 안팎의 초단거리에서 저전력 무선 연결이 필요할 때 쓰인다.
- 블루투스의 무선 시스템은 ISM(Industrial Scientific and Medical) 주파수 대역인 2400~2483.5MHz를 사용한다. 이 중 위아래 주파수를 쓰는 다른 시스템들의 간섭을 막기 위해 2400MHz 이후 2MHz, 2483.5MHz 이전 3.5MHz까지의 범위를 제외한 2402~2480MHz, 총 79개 채널을 쓴다. ISM이란 산업, 과학, 의료용으로 할당된 주파수 대역으로, 전파 사용에 대해 허가를 받을 필요가 없어 저전력의 전파를 발산하는 개인 무선기기에 많이 쓰인다. 아마추어 무선, 무선랜, 블루투스가 이 ISM 대역을 사용한다.

○ 무선 통신을 이용한 연결 인터페이스 구축

- 무선 통신을 이용한 연결 인터페이스 구축을 위해 다음과 같은 과정을 통해서 개발하였다.
 - 단위 기능 분석 단계
 - ① 진동 측정 모듈 설계 및 구현
 - ② 최대/최소 허용값 제한 기능 구현
 - 무선 모듈 구현 및 적용 단계
 - ① 블루투스를 이용한 무선 모듈 적용
 - 연동 모듈 개발 및 확인 단계
 - ① HW 와 SW(앱 시스템)와의 통신 프로토콜 구현

② 송수신 정보 단위별 확인 실시(1차 터미널 프로그램 이용)

- 필드 테스트 단계

- ① 실내 단위 테스트 실시 : 임의의 진동 발생 후 데이터 전송 및 DB 저장 루틴 확인 (정상)
- ② 시험발파 현장 테스트 실시 : 시험발파 현장에서 실제 발파에 따른 진동 측정값 송수신 및 DB 저장 루틴 확인 (정상)

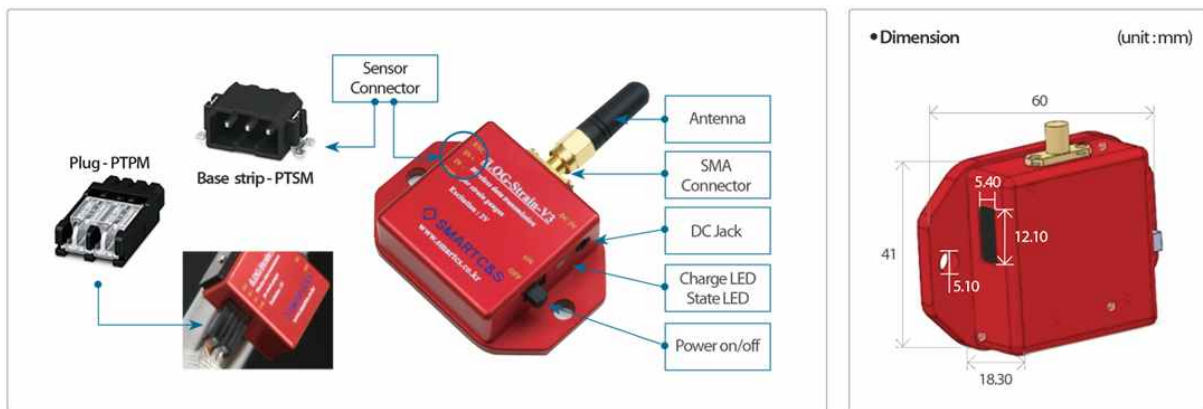
○ 무선계측기와 연동하여 실시간 정보처리가 가능한 IoT(사물인터넷, Internet of Things) 기반의 연결 인터페이스 구축을 위한 시스템 구성



[시스템 구성도]

- 현장 측정을 위한 소형화 및 무선 모듈 탑재

- 블루투스 통신 프로토콜 적용



[진동 계측기 구성도]

- 무선 통신용 계측기와의 인터페이스 연결을 위한 블루투스 SW 모듈 개발

① Bluetooth Class Method 분석

- BluetoothAdapter

Represents the local Bluetooth adapter (Bluetooth radio). The BluetoothAdapter is the entry-point for all Bluetooth interaction. Using this, you can discover other Bluetooth devices, query a list of bonded (paired) devices, instantiate a BluetoothDevice using a known MAC address, and create a BluetoothServerSocket to listen for communications from other devices.

- BluetoothDevice

Represents a remote Bluetooth device. Use this to request a connection with a remote device through a BluetoothSocket or query information about the device such as its name, address, class, and bonding state.

- BluetoothSocket

Represents the interface for a Bluetooth socket (similar to a TCP Socket). This is the connection point that allows an application to exchange data with another Bluetooth device via InputStream and OutputStream.

- BluetoothServerSocket

Represents an open server socket that listens for incoming requests (similar to a TCP ServerSocket). In order to connect two Android devices, one device must open a server socket with this class. When a remote Bluetooth device makes a connection request to the this device, the BluetoothServerSocket will return a connected BluetoothSocket when the connection is accepted.

- BluetoothClass

Describes the general characteristics and capabilities of a Bluetooth device. This is a read-only set of properties that define the device's major and minor device classes and its services. However, this does not reliably describe all Bluetooth profiles and services supported by the device, but is useful as a hint to the device type.

- BluetoothProfile

An interface that represents a Bluetooth profile. A Bluetooth profile is a wireless interface specification for Bluetooth-based communication between devices. An example is the Hands-Free profile. For more discussion of profiles, see Working with Profiles

- BluetoothHeadset

Provides support for Bluetooth headsets to be used with mobile phones. This includes

both Bluetooth Headset and Hands-Free (v1.5) profiles.

- BluetoothA2dp

Defines how high quality audio can be streamed from one device to another over a Bluetooth connection. "A2DP" stands for Advanced Audio Distribution Profile.

- BluetoothHealth

Represents a Health Device Profile proxy that controls the Bluetooth service.

BluetoothHealthCallback

An abstract class that you use to implement BluetoothHealth callbacks. You must extend this class and implement the callback methods to receive updates about changes in the application's registration state and Bluetooth channel state.

- BluetoothHealthAppConfiguration

Represents an application configuration that the Bluetooth Health third-party application registers to communicate with a remote Bluetooth health device.

- BluetoothProfile.ServiceListener

An interface that notifies BluetoothProfile IPC clients when they have been connected to or disconnected from the service (that is, the internal service that runs a particular profile).

② Bluetooth Class 구현 방법

- Bluetooth Permissions 설정

```
<manifest ... >
  <uses-permission android:name="android.permission.BLUETOOTH" />
  ...
</manifest>
```

- Get the BluetoothAdapter

The BluetoothAdapter is required for any and all Bluetooth activity. To get the BluetoothAdapter, call the static getDefaultAdapter() method. This returns a BluetoothAdapter that represents the device's own Bluetooth adapter (the Bluetooth radio). There's one Bluetooth adapter for the entire system, and your application can interact with it using this object. If getDefaultAdapter() returns null, then the device does not support Bluetooth and your story ends here

```
BluetoothAdapter mBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
if (mBluetoothAdapter == null) {
    // Device does not support Bluetooth
}
```

- Enable Bluetooth

Next, you need to ensure that Bluetooth is enabled. Call isEnabled() to check whether Bluetooth is currently enable. If this method returns false, then Bluetooth is

disabled. To request that Bluetooth be enabled, call `startActivityForResult()` with the `ACTION_REQUEST_ENABLE` action Intent. This will issue a request to enable Bluetooth through the system settings (without stopping your application).

```
if (!BluetoothAdapter.isEnabled()) {
    Intent enableBtIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
    startActivityForResult(enableBtIntent, REQUEST_ENABLE_BT);
}
```

- Querying paired devices

Before performing device discovery, it's worth querying the set of paired devices to see if the desired device is already known. To do so, call `getBondedDevices()`. This will return a Set of `BluetoothDevices` representing paired devices. For example, you can query all paired devices and then show the name of each device to the user, using an `ArrayAdapter`

```
Set<BluetoothDevice> pairedDevices = mBluetoothAdapter.getBondedDevices();
// If there are paired devices
if (pairedDevices.size() > 0) {
    // Loop through paired devices
    for (BluetoothDevice device : pairedDevices) {
        // Add the name and address to an array adapter to show in a ListView
        mAdapter.add(device.getName() + "\n" + device.getAddress());
    }
}
```

- Discovering devices

To start discovering devices, simply call `startDiscovery()`. The process is asynchronous and the method will immediately return with a boolean indicating whether discovery has successfully started. The discovery process usually involves an inquiry scan of about 12 seconds, followed by a page scan of each found device to retrieve its Bluetooth name.

```
// Create a BroadcastReceiver for ACTION_FOUND
private final BroadcastReceiver mReceiver = new BroadcastReceiver() {
    public void onReceive(Context context, Intent intent) {
        String action = intent.getAction();
        // When discovery finds a device
        if (BluetoothDevice.ACTION_FOUND.equals(action)) {
            // Get the BluetoothDevice object from the Intent
            BluetoothDevice device = intent.getParcelableExtra(BluetoothDevice.EXTRA_DEVICE);
            // Add the name and address to an array adapter to show in a ListView
            mAdapter.add(device.getName() + "\n" + device.getAddress());
        }
    }
};
// Register the BroadcastReceiver
IntentFilter filter = new IntentFilter(BluetoothDevice.ACTION_FOUND);
registerReceiver(mReceiver, filter); // Don't forget to unregister during onDestroy
```

- Enabling discoverability

If you would like to make the local device discoverable to other devices, call `startActivityForResult(Intent, int)` with the `ACTION_REQUEST_DISCOVERABLE` action Intent. This will issue a request to enable discoverable mode through the system settings (without stopping your application). By default, the device will become discoverable for 120 seconds. You can define a different duration by adding the `EXTRA_DISCOVERABLE_DURATION` Intent extra. The maximum duration an app can set is 3600 seconds, and a value of 0 means the device is always discoverable. Any value below 0 or

above 3600 is automatically set to 120 secs). For example, this snippet sets the duration to 300

```
Intent discoverableIntent = new
Intent(BluetoothAdapter.ACTION_REQUEST_DISCOVERABLE);
discoverableIntent.putExtra(BluetoothAdapter.EXTRA_DISCOVERABLE_DURATION, 300);
startActivity(discoverableIntent);
```

- Connecting as a server

When you want to connect two devices, one must act as a server by holding an open `BluetoothServerSocket`. The purpose of the server socket is to listen for incoming connection requests and when one is accepted, provide a connected `BluetoothSocket`. When the `BluetoothSocket` is acquired from the `BluetoothServerSocket`, the `BluetoothServerSocket` can (and should) be discarded, unless you want to accept more connections.

```
private class AcceptThread extends Thread {
    private final BluetoothServerSocket mmServerSocket;

    public AcceptThread() {
        // Use a temporary object that is later assigned to mmServerSocket,
        // because mmServerSocket is final
        BluetoothServerSocket tmp = null;
        try {
            // MY_UUID is the app's UUID string, also used by the client code
            tmp = mBluetoothAdapter.listenUsingRfcommWithServiceRecord(NAME, MY_UUID);
        } catch (IOException e) { }
        mmServerSocket = tmp;
    }

    public void run() {
        BluetoothSocket socket = null;
        // Keep listening until exception occurs or a socket is returned
        while (true) {
            try {
                socket = mmServerSocket.accept();
            } catch (IOException e) {
                break;
            }
            // If a connection was accepted
            if (socket != null) {
                // Do work to manage the connection (in a separate thread)
                manageConnectedSocket(socket);
                mmServerSocket.close();
                break;
            }
        }
    }

    /** Will cancel the listening socket, and cause the thread to finish */
    public void cancel() {
        try {
            mmServerSocket.close();
        } catch (IOException e) { }
    }
}
```

- Connecting as a client

In order to initiate a connection with a remote device (a device holding an open server socket), you must first obtain a `BluetoothDevice` object that represents the remote device. (Getting a `BluetoothDevice` is covered in the above section about Finding Devices.) You must then use the `BluetoothDevice` to acquire a `BluetoothSocket` and initiate the connection.

```

private class ConnectThread extends Thread {
    private final BluetoothSocket mmSocket;
    private final BluetoothDevice mmDevice;

    public ConnectThread(BluetoothDevice device) {
        // Use a temporary object that is later assigned to mmSocket,
        // because mmSocket is final
        BluetoothSocket tmp = null;
        mmDevice = device;

        // Get a BluetoothSocket to connect with the given BluetoothDevice
        try {
            // MY_UUID is the app's UUID string, also used by the server code
            tmp = device.createRfcommSocketToServiceRecord(MY_UUID);
        } catch (IOException e) { }
        mmSocket = tmp;
    }

    public void run() {
        // Cancel discovery because it will slow down the connection
        mBluetoothAdapter.cancelDiscovery();

        try {
            // Connect the device through the socket. This will block
            // until it succeeds or throws an exception
            mmSocket.connect();
        } catch (IOException connectException) {
            // Unable to connect; close the socket and get out
            try {
                mmSocket.close();
            } catch (IOException closeException) { }
            return;
        }

        // Do work to manage the connection (in a separate thread)
        manageConnectedSocket(mmSocket);
    }

    /** Will cancel an in-progress connection, and close the socket */
    public void cancel() {
        try {
            mmSocket.close();
        } catch (IOException e) { }
    }
}

```

- Managing a Connection

When you have successfully connected two (or more) devices, each one will have a connected `BluetoothSocket`. This is where the fun begins because you can share data between devices. Using the `BluetoothSocket`, the general procedure to transfer arbitrary data is simple

There are, of course, implementation details to consider. First and foremost, you should use a dedicated thread for all stream reading and writing. This is important because both `read(byte[])` and `write(byte[])` methods are blocking calls. `read(byte[])` will block until there is something to read from the stream. `write(byte[])` does not usually block, but can block for flow control if the remote device is not calling `read(byte[])` quickly enough and the intermediate buffers are full. So, your main loop in the thread should be dedicated to reading from the `InputStream`. A separate public method in the thread can be used to initiate writes to the `OutputStream`.

```

private class ConnectedThread extends Thread {
    private final BluetoothSocket mmSocket;
    private final InputStream mmInStream;
    private final OutputStream mmOutStream;

    public ConnectedThread(BluetoothSocket socket) {
        mmSocket = socket;
        InputStream tmpIn = null;
        OutputStream tmpOut = null;

        // Get the input and output streams, using temp objects because
        // member streams are final
        try {
            tmpIn = socket.getInputStream();
            tmpOut = socket.getOutputStream();
        } catch (IOException e) { }

        mmInStream = tmpIn;
        mmOutStream = tmpOut;
    }

    public void run() {
        byte[] buffer = new byte[1024]; // buffer store for the stream
        int bytes; // bytes returned from read()

        // Keep listening to the InputStream until an exception occurs
        while (true) {
            try {
                // Read from the InputStream
                bytes = mmInStream.read(buffer);
                // Send the obtained bytes to the UI activity
                mHandler.obtainMessage(MESSAGE_READ, bytes, -1, buffer)
                    .sendToTarget();
            } catch (IOException e) {
                break;
            }
        }
    }

    /* Call this from the main activity to send data to the remote device */
    public void write(byte[] bytes) {
        try {
            mmOutStream.write(bytes);
        } catch (IOException e) { }
    }

    /* Call this from the main activity to shutdown the connection */
    public void cancel() {
        try {
            mmSocket.close();
        } catch (IOException e) { }
    }
}

```

- 계측기와의 무선 인터페이스 연결 결과

- 계측센서 데이터 정보 수신 여부를 실시하였다

- 다음과 같이 5회 실시하여 정보를 수집하는 데 이상이 없었다.

시험발파명 KGI 시험발파 추가 속정거리 차수 Search Excel Highest Value X: 4.07 cm/sec, Y: 3.07 cm/sec, Z: 3.53 cm/sec <table> <tr> <th>No</th><th>거리</th><th>차수</th><th>X</th><th>Y</th><th>Z</th><th>측정일시</th></tr> <tr><td>455</td><td>20m</td><td>1차</td><td>2.8 cm/sec</td><td>2.39 cm/sec</td><td>2.75 cm/sec</td><td>2018-12-10 10:53:03</td></tr> <tr><td>454</td><td>20m</td><td>1차</td><td>2.81 cm/sec</td><td>2.39 cm/sec</td><td>2.75 cm/sec</td><td>2018-12-10 10:53:03</td></tr> <tr><td>453</td><td>20m</td><td>1차</td><td>2.79 cm/sec</td><td>2.41 cm/sec</td><td>2.75 cm/sec</td><td>2018-12-10 10:53:03</td></tr> <tr><td>452</td><td>20m</td><td>1차</td><td>2.75 cm/sec</td><td>2.44 cm/sec</td><td>2.75 cm/sec</td><td>2018-12-10 10:53:03</td></tr> <tr><td>451</td><td>20m</td><td>1차</td><td>2.74 cm/sec</td><td>2.45 cm/sec</td><td>2.75 cm/sec</td><td>2018-12-10 10:53:03</td></tr> <tr><td>450</td><td>20m</td><td>1차</td><td>2.82 cm/sec</td><td>2.4 cm/sec</td><td>2.75 cm/sec</td><td>2018-12-10 10:53:03</td></tr> <tr><td>449</td><td>20m</td><td>1차</td><td>2.78 cm/sec</td><td>2.42 cm/sec</td><td>2.75 cm/sec</td><td>2018-12-10 10:53:03</td></tr> <tr><td>448</td><td>20m</td><td>1차</td><td>2.72 cm/sec</td><td>2.43 cm/sec</td><td>2.75 cm/sec</td><td>2018-12-10 10:53:03</td></tr> <tr><td>447</td><td>20m</td><td>1차</td><td>2.65 cm/sec</td><td>2.45 cm/sec</td><td>2.76 cm/sec</td><td>2018-12-10 10:53:03</td></tr> <tr><td>446</td><td>20m</td><td>1차</td><td>2.59 cm/sec</td><td>2.46 cm/sec</td><td>2.76 cm/sec</td><td>2018-12-10 10:53:03</td></tr> </table>							No	거리	차수	X	Y	Z	측정일시	455	20m	1차	2.8 cm/sec	2.39 cm/sec	2.75 cm/sec	2018-12-10 10:53:03	454	20m	1차	2.81 cm/sec	2.39 cm/sec	2.75 cm/sec	2018-12-10 10:53:03	453	20m	1차	2.79 cm/sec	2.41 cm/sec	2.75 cm/sec	2018-12-10 10:53:03	452	20m	1차	2.75 cm/sec	2.44 cm/sec	2.75 cm/sec	2018-12-10 10:53:03	451	20m	1차	2.74 cm/sec	2.45 cm/sec	2.75 cm/sec	2018-12-10 10:53:03	450	20m	1차	2.82 cm/sec	2.4 cm/sec	2.75 cm/sec	2018-12-10 10:53:03	449	20m	1차	2.78 cm/sec	2.42 cm/sec	2.75 cm/sec	2018-12-10 10:53:03	448	20m	1차	2.72 cm/sec	2.43 cm/sec	2.75 cm/sec	2018-12-10 10:53:03	447	20m	1차	2.65 cm/sec	2.45 cm/sec	2.76 cm/sec	2018-12-10 10:53:03	446	20m	1차	2.59 cm/sec	2.46 cm/sec	2.76 cm/sec	2018-12-10 10:53:03																					
No	거리	차수	X	Y	Z	측정일시																																																																																																		
455	20m	1차	2.8 cm/sec	2.39 cm/sec	2.75 cm/sec	2018-12-10 10:53:03																																																																																																		
454	20m	1차	2.81 cm/sec	2.39 cm/sec	2.75 cm/sec	2018-12-10 10:53:03																																																																																																		
453	20m	1차	2.79 cm/sec	2.41 cm/sec	2.75 cm/sec	2018-12-10 10:53:03																																																																																																		
452	20m	1차	2.75 cm/sec	2.44 cm/sec	2.75 cm/sec	2018-12-10 10:53:03																																																																																																		
451	20m	1차	2.74 cm/sec	2.45 cm/sec	2.75 cm/sec	2018-12-10 10:53:03																																																																																																		
450	20m	1차	2.82 cm/sec	2.4 cm/sec	2.75 cm/sec	2018-12-10 10:53:03																																																																																																		
449	20m	1차	2.78 cm/sec	2.42 cm/sec	2.75 cm/sec	2018-12-10 10:53:03																																																																																																		
448	20m	1차	2.72 cm/sec	2.43 cm/sec	2.75 cm/sec	2018-12-10 10:53:03																																																																																																		
447	20m	1차	2.65 cm/sec	2.45 cm/sec	2.76 cm/sec	2018-12-10 10:53:03																																																																																																		
446	20m	1차	2.59 cm/sec	2.46 cm/sec	2.76 cm/sec	2018-12-10 10:53:03																																																																																																		
시험발파명 KGI 시험발파 추가 속정거리 차수 Search Excel Highest Value X: 4.07 cm/sec, Y: 3.07 cm/sec, Z: 3.53 cm/sec <table> <tr> <th>No</th><th>거리</th><th>차수</th><th>X</th><th>Y</th><th>Z</th><th>측정일시</th></tr> <tr><td>905</td><td>20m</td><td>2차</td><td>2.32 cm/sec</td><td>2.57 cm/sec</td><td>2.75 cm/sec</td><td>2018-12-11 10:53:43</td></tr> <tr><td>904</td><td>20m</td><td>2차</td><td>2.5 cm/sec</td><td>2.42 cm/sec</td><td>2.77 cm/sec</td><td>2018-12-11 10:53:43</td></tr> <tr><td>903</td><td>20m</td><td>2차</td><td>2.42 cm/sec</td><td>2.44 cm/sec</td><td>2.76 cm/sec</td><td>2018-12-11 10:53:43</td></tr> <tr><td>902</td><td>20m</td><td>2차</td><td>2.37 cm/sec</td><td>2.45 cm/sec</td><td>2.75 cm/sec</td><td>2018-12-11 10:53:43</td></tr> <tr><td>901</td><td>20m</td><td>2차</td><td>2.38 cm/sec</td><td>2.45 cm/sec</td><td>2.76 cm/sec</td><td>2018-12-11 10:53:43</td></tr> <tr><td>900</td><td>20m</td><td>2차</td><td>2.37 cm/sec</td><td>2.39 cm/sec</td><td>2.76 cm/sec</td><td>2018-12-11 10:53:43</td></tr> <tr><td>899</td><td>20m</td><td>2차</td><td>2.27 cm/sec</td><td>2.44 cm/sec</td><td>2.75 cm/sec</td><td>2018-12-11 10:53:43</td></tr> <tr><td>898</td><td>20m</td><td>2차</td><td>2.27 cm/sec</td><td>2.46 cm/sec</td><td>2.76 cm/sec</td><td>2018-12-11 10:53:43</td></tr> <tr><td>897</td><td>20m</td><td>2차</td><td>2.28 cm/sec</td><td>2.46 cm/sec</td><td>2.75 cm/sec</td><td>2018-12-11 10:53:43</td></tr> <tr><td>896</td><td>20m</td><td>2차</td><td>2.3 cm/sec</td><td>2.45 cm/sec</td><td>2.75 cm/sec</td><td>2018-12-11 10:53:43</td></tr> <tr><td>895</td><td>20m</td><td>2차</td><td>2.31 cm/sec</td><td>2.4 cm/sec</td><td>2.75 cm/sec</td><td>2018-12-11 10:53:43</td></tr> <tr><td>894</td><td>20m</td><td>2차</td><td>2.3 cm/sec</td><td>2.37 cm/sec</td><td>2.76 cm/sec</td><td>2018-12-11 10:53:43</td></tr> <tr><td>893</td><td>20m</td><td>2차</td><td>2.34 cm/sec</td><td>2.38 cm/sec</td><td>2.76 cm/sec</td><td>2018-12-11 10:53:43</td></tr> </table>							No	거리	차수	X	Y	Z	측정일시	905	20m	2차	2.32 cm/sec	2.57 cm/sec	2.75 cm/sec	2018-12-11 10:53:43	904	20m	2차	2.5 cm/sec	2.42 cm/sec	2.77 cm/sec	2018-12-11 10:53:43	903	20m	2차	2.42 cm/sec	2.44 cm/sec	2.76 cm/sec	2018-12-11 10:53:43	902	20m	2차	2.37 cm/sec	2.45 cm/sec	2.75 cm/sec	2018-12-11 10:53:43	901	20m	2차	2.38 cm/sec	2.45 cm/sec	2.76 cm/sec	2018-12-11 10:53:43	900	20m	2차	2.37 cm/sec	2.39 cm/sec	2.76 cm/sec	2018-12-11 10:53:43	899	20m	2차	2.27 cm/sec	2.44 cm/sec	2.75 cm/sec	2018-12-11 10:53:43	898	20m	2차	2.27 cm/sec	2.46 cm/sec	2.76 cm/sec	2018-12-11 10:53:43	897	20m	2차	2.28 cm/sec	2.46 cm/sec	2.75 cm/sec	2018-12-11 10:53:43	896	20m	2차	2.3 cm/sec	2.45 cm/sec	2.75 cm/sec	2018-12-11 10:53:43	895	20m	2차	2.31 cm/sec	2.4 cm/sec	2.75 cm/sec	2018-12-11 10:53:43	894	20m	2차	2.3 cm/sec	2.37 cm/sec	2.76 cm/sec	2018-12-11 10:53:43	893	20m	2차	2.34 cm/sec	2.38 cm/sec	2.76 cm/sec	2018-12-11 10:53:43
No	거리	차수	X	Y	Z	측정일시																																																																																																		
905	20m	2차	2.32 cm/sec	2.57 cm/sec	2.75 cm/sec	2018-12-11 10:53:43																																																																																																		
904	20m	2차	2.5 cm/sec	2.42 cm/sec	2.77 cm/sec	2018-12-11 10:53:43																																																																																																		
903	20m	2차	2.42 cm/sec	2.44 cm/sec	2.76 cm/sec	2018-12-11 10:53:43																																																																																																		
902	20m	2차	2.37 cm/sec	2.45 cm/sec	2.75 cm/sec	2018-12-11 10:53:43																																																																																																		
901	20m	2차	2.38 cm/sec	2.45 cm/sec	2.76 cm/sec	2018-12-11 10:53:43																																																																																																		
900	20m	2차	2.37 cm/sec	2.39 cm/sec	2.76 cm/sec	2018-12-11 10:53:43																																																																																																		
899	20m	2차	2.27 cm/sec	2.44 cm/sec	2.75 cm/sec	2018-12-11 10:53:43																																																																																																		
898	20m	2차	2.27 cm/sec	2.46 cm/sec	2.76 cm/sec	2018-12-11 10:53:43																																																																																																		
897	20m	2차	2.28 cm/sec	2.46 cm/sec	2.75 cm/sec	2018-12-11 10:53:43																																																																																																		
896	20m	2차	2.3 cm/sec	2.45 cm/sec	2.75 cm/sec	2018-12-11 10:53:43																																																																																																		
895	20m	2차	2.31 cm/sec	2.4 cm/sec	2.75 cm/sec	2018-12-11 10:53:43																																																																																																		
894	20m	2차	2.3 cm/sec	2.37 cm/sec	2.76 cm/sec	2018-12-11 10:53:43																																																																																																		
893	20m	2차	2.34 cm/sec	2.38 cm/sec	2.76 cm/sec	2018-12-11 10:53:43																																																																																																		
시험발파명 KGI 시험발파 추가 속정거리 차수 Search Excel Highest Value X: 4.07 cm/sec, Y: 3.07 cm/sec, Z: 3.53 cm/sec <table> <tr> <th>No</th><th>거리</th><th>차수</th><th>X</th><th>Y</th><th>Z</th><th>측정일시</th></tr> <tr><td>2405</td><td>20m</td><td>3차</td><td>1.92 cm/sec</td><td>2.57 cm/sec</td><td>2.78 cm/sec</td><td>2018-12-12 11:30:24</td></tr> <tr><td>2404</td><td>20m</td><td>3차</td><td>1.73 cm/sec</td><td>2.51 cm/sec</td><td>2.77 cm/sec</td><td>2018-12-12 11:30:24</td></tr> <tr><td>2403</td><td>20m</td><td>3차</td><td>1.58 cm/sec</td><td>2.52 cm/sec</td><td>2.77 cm/sec</td><td>2018-12-12 11:30:24</td></tr> <tr><td>2402</td><td>20m</td><td>3차</td><td>1.49 cm/sec</td><td>2.46 cm/sec</td><td>2.77 cm/sec</td><td>2018-12-12 11:30:24</td></tr> <tr><td>2401</td><td>20m</td><td>3차</td><td>1.39 cm/sec</td><td>2.46 cm/sec</td><td>2.76 cm/sec</td><td>2018-12-12 11:30:24</td></tr> <tr><td>2400</td><td>20m</td><td>3차</td><td>1.34 cm/sec</td><td>2.51 cm/sec</td><td>2.76 cm/sec</td><td>2018-12-12 11:30:24</td></tr> <tr><td>2399</td><td>20m</td><td>3차</td><td>1.54 cm/sec</td><td>2.51 cm/sec</td><td>2.76 cm/sec</td><td>2018-12-12 11:30:24</td></tr> <tr><td>2398</td><td>20m</td><td>3차</td><td>1.78 cm/sec</td><td>2.5 cm/sec</td><td>2.76 cm/sec</td><td>2018-12-12 11:30:24</td></tr> <tr><td>2397</td><td>20m</td><td>3차</td><td>2.08 cm/sec</td><td>2.48 cm/sec</td><td>2.75 cm/sec</td><td>2018-12-12 11:30:24</td></tr> <tr><td>2396</td><td>20m</td><td>3차</td><td>2.25 cm/sec</td><td>2.48 cm/sec</td><td>2.78 cm/sec</td><td>2018-12-12 11:30:24</td></tr> </table>							No	거리	차수	X	Y	Z	측정일시	2405	20m	3차	1.92 cm/sec	2.57 cm/sec	2.78 cm/sec	2018-12-12 11:30:24	2404	20m	3차	1.73 cm/sec	2.51 cm/sec	2.77 cm/sec	2018-12-12 11:30:24	2403	20m	3차	1.58 cm/sec	2.52 cm/sec	2.77 cm/sec	2018-12-12 11:30:24	2402	20m	3차	1.49 cm/sec	2.46 cm/sec	2.77 cm/sec	2018-12-12 11:30:24	2401	20m	3차	1.39 cm/sec	2.46 cm/sec	2.76 cm/sec	2018-12-12 11:30:24	2400	20m	3차	1.34 cm/sec	2.51 cm/sec	2.76 cm/sec	2018-12-12 11:30:24	2399	20m	3차	1.54 cm/sec	2.51 cm/sec	2.76 cm/sec	2018-12-12 11:30:24	2398	20m	3차	1.78 cm/sec	2.5 cm/sec	2.76 cm/sec	2018-12-12 11:30:24	2397	20m	3차	2.08 cm/sec	2.48 cm/sec	2.75 cm/sec	2018-12-12 11:30:24	2396	20m	3차	2.25 cm/sec	2.48 cm/sec	2.78 cm/sec	2018-12-12 11:30:24																					
No	거리	차수	X	Y	Z	측정일시																																																																																																		
2405	20m	3차	1.92 cm/sec	2.57 cm/sec	2.78 cm/sec	2018-12-12 11:30:24																																																																																																		
2404	20m	3차	1.73 cm/sec	2.51 cm/sec	2.77 cm/sec	2018-12-12 11:30:24																																																																																																		
2403	20m	3차	1.58 cm/sec	2.52 cm/sec	2.77 cm/sec	2018-12-12 11:30:24																																																																																																		
2402	20m	3차	1.49 cm/sec	2.46 cm/sec	2.77 cm/sec	2018-12-12 11:30:24																																																																																																		
2401	20m	3차	1.39 cm/sec	2.46 cm/sec	2.76 cm/sec	2018-12-12 11:30:24																																																																																																		
2400	20m	3차	1.34 cm/sec	2.51 cm/sec	2.76 cm/sec	2018-12-12 11:30:24																																																																																																		
2399	20m	3차	1.54 cm/sec	2.51 cm/sec	2.76 cm/sec	2018-12-12 11:30:24																																																																																																		
2398	20m	3차	1.78 cm/sec	2.5 cm/sec	2.76 cm/sec	2018-12-12 11:30:24																																																																																																		
2397	20m	3차	2.08 cm/sec	2.48 cm/sec	2.75 cm/sec	2018-12-12 11:30:24																																																																																																		
2396	20m	3차	2.25 cm/sec	2.48 cm/sec	2.78 cm/sec	2018-12-12 11:30:24																																																																																																		
시험발파명 KGI 시험발파3 추가 속정거리 차수 4 Search Excel Highest Value X: 2.60 cm/sec, Y: 2.60 cm/sec, Z: 2.60 cm/sec <table> <tr> <th>No</th><th>거리</th><th>차수</th><th>X</th><th>Y</th><th>Z</th><th>측정일시</th></tr> <tr><td>2</td><td>40m</td><td>4차</td><td>2.6 cm/sec</td><td>2.6 cm/sec</td><td>2.6 cm/sec</td><td>2018-12-13 5:01:35</td></tr> <tr><td>1</td><td>40m</td><td>4차</td><td>2.5 cm/sec</td><td>2.5 cm/sec</td><td>2.5 cm/sec</td><td>2018-12-13 5:01:35</td></tr> </table>							No	거리	차수	X	Y	Z	측정일시	2	40m	4차	2.6 cm/sec	2.6 cm/sec	2.6 cm/sec	2018-12-13 5:01:35	1	40m	4차	2.5 cm/sec	2.5 cm/sec	2.5 cm/sec	2018-12-13 5:01:35																																																																													
No	거리	차수	X	Y	Z	측정일시																																																																																																		
2	40m	4차	2.6 cm/sec	2.6 cm/sec	2.6 cm/sec	2018-12-13 5:01:35																																																																																																		
1	40m	4차	2.5 cm/sec	2.5 cm/sec	2.5 cm/sec	2018-12-13 5:01:35																																																																																																		
시험발파명 KGI 시험발파 추가 속정거리 차수 6 Search Excel Highest Value X: 4.07 cm/sec, Y: 3.07 cm/sec, Z: 3.53 cm/sec <table> <tr> <th>No</th><th>거리</th><th>차수</th><th>X</th><th>Y</th><th>Z</th><th>측정일시</th></tr> <tr><td>861</td><td>40m</td><td>6차</td><td>2.42 cm/sec</td><td>2.42 cm/sec</td><td>2.75 cm/sec</td><td>2018-12-14 11:58:36</td></tr> <tr><td>860</td><td>40m</td><td>6차</td><td>2.43 cm/sec</td><td>2.43 cm/sec</td><td>2.75 cm/sec</td><td>2018-12-14 11:58:36</td></tr> <tr><td>859</td><td>40m</td><td>6차</td><td>2.42 cm/sec</td><td>2.43 cm/sec</td><td>2.75 cm/sec</td><td>2018-12-14 11:58:36</td></tr> <tr><td>858</td><td>40m</td><td>6차</td><td>2.42 cm/sec</td><td>2.43 cm/sec</td><td>2.75 cm/sec</td><td>2018-12-14 11:58:36</td></tr> <tr><td>857</td><td>40m</td><td>6차</td><td>2.42 cm/sec</td><td>2.43 cm/sec</td><td>2.75 cm/sec</td><td>2018-12-14 11:58:36</td></tr> <tr><td>856</td><td>40m</td><td>6차</td><td>2.43 cm/sec</td><td>2.43 cm/sec</td><td>2.75 cm/sec</td><td>2018-12-14 11:58:36</td></tr> <tr><td>855</td><td>40m</td><td>6차</td><td>2.42 cm/sec</td><td>2.43 cm/sec</td><td>2.75 cm/sec</td><td>2018-12-14 11:58:36</td></tr> <tr><td>854</td><td>40m</td><td>6차</td><td>2.42 cm/sec</td><td>2.43 cm/sec</td><td>2.76 cm/sec</td><td>2018-12-14 11:58:36</td></tr> </table>							No	거리	차수	X	Y	Z	측정일시	861	40m	6차	2.42 cm/sec	2.42 cm/sec	2.75 cm/sec	2018-12-14 11:58:36	860	40m	6차	2.43 cm/sec	2.43 cm/sec	2.75 cm/sec	2018-12-14 11:58:36	859	40m	6차	2.42 cm/sec	2.43 cm/sec	2.75 cm/sec	2018-12-14 11:58:36	858	40m	6차	2.42 cm/sec	2.43 cm/sec	2.75 cm/sec	2018-12-14 11:58:36	857	40m	6차	2.42 cm/sec	2.43 cm/sec	2.75 cm/sec	2018-12-14 11:58:36	856	40m	6차	2.43 cm/sec	2.43 cm/sec	2.75 cm/sec	2018-12-14 11:58:36	855	40m	6차	2.42 cm/sec	2.43 cm/sec	2.75 cm/sec	2018-12-14 11:58:36	854	40m	6차	2.42 cm/sec	2.43 cm/sec	2.76 cm/sec	2018-12-14 11:58:36																																			
No	거리	차수	X	Y	Z	측정일시																																																																																																		
861	40m	6차	2.42 cm/sec	2.42 cm/sec	2.75 cm/sec	2018-12-14 11:58:36																																																																																																		
860	40m	6차	2.43 cm/sec	2.43 cm/sec	2.75 cm/sec	2018-12-14 11:58:36																																																																																																		
859	40m	6차	2.42 cm/sec	2.43 cm/sec	2.75 cm/sec	2018-12-14 11:58:36																																																																																																		
858	40m	6차	2.42 cm/sec	2.43 cm/sec	2.75 cm/sec	2018-12-14 11:58:36																																																																																																		
857	40m	6차	2.42 cm/sec	2.43 cm/sec	2.75 cm/sec	2018-12-14 11:58:36																																																																																																		
856	40m	6차	2.43 cm/sec	2.43 cm/sec	2.75 cm/sec	2018-12-14 11:58:36																																																																																																		
855	40m	6차	2.42 cm/sec	2.43 cm/sec	2.75 cm/sec	2018-12-14 11:58:36																																																																																																		
854	40m	6차	2.42 cm/sec	2.43 cm/sec	2.76 cm/sec	2018-12-14 11:58:36																																																																																																		

3. 시설물 3D 정보 연동 인터페이스 모듈 구현

(1) 시설물 3D 정보 연동 인터페이스 모듈의 개요

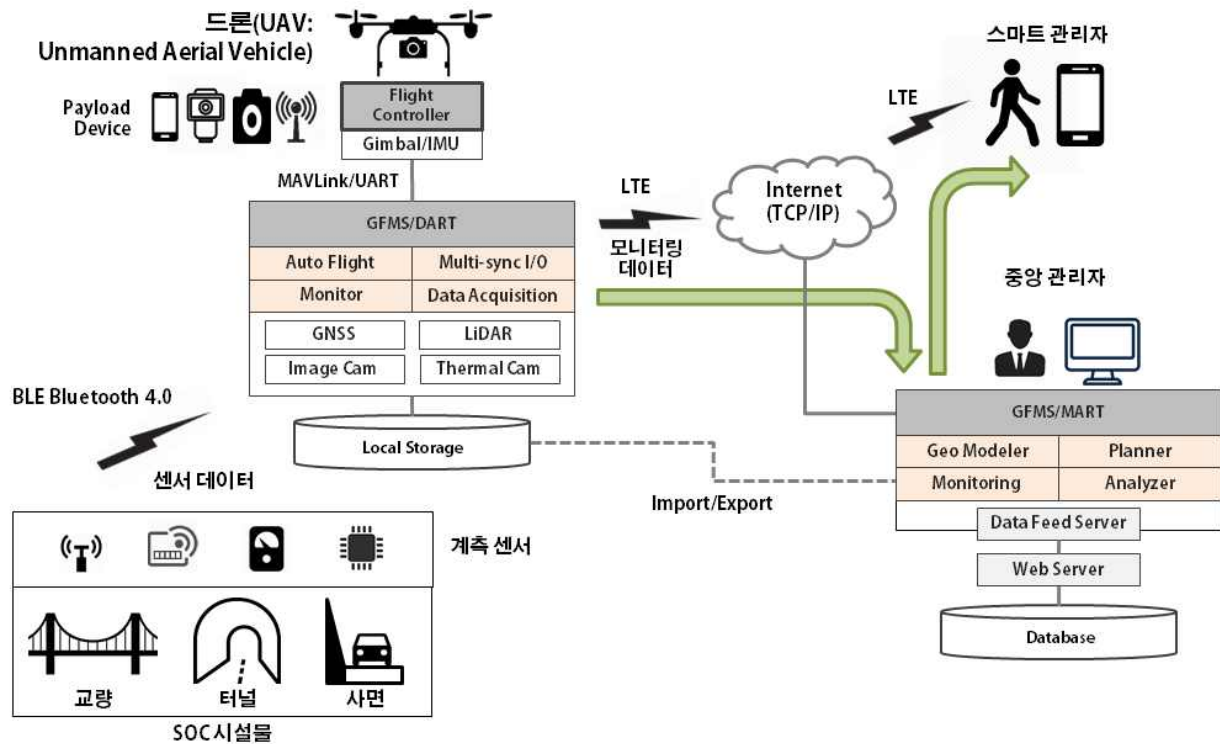
시설물 및 지형 변환의 분석은 위험 요소 모니터링을 위한 기초 데이터로서 데이터의 정밀도 및 결과에 따라 위험을 예측하는 중요한 요소이다. 정확한 지형 변환량을 분석하기 위한 시스템 요소들은 다음과 같다.

- ① 모니터링이 수행되는 지형의 형태를 정밀하게 표현할 수 있는 공간 정보
- ② 지형 변환량 계산과정의 적용
- ③ 효율적 위험 예측 및 관리 계획의 수립
- ④ 시각화 및 정보 전파 체계

따라서 현재 이러한 요소들을 고려하여 정확한 지형 변환량을 계산하고, 효율적 재난관리 계획을 수립하기 위한 다양한 연구가 수행되고 있다. 최근에는 지형 변환량 계산을 위한 정밀한 지형자료의 취득을 위해 항공 사진, 라이다, 3D 레이저스캐닝, GPS 등을 활용한 지리정보시스템(GIS)과 같은 최신의 측량기술 및 자료처리 기술을 이용하고 있는데, 이 중 항공 라이다에 대한 연구가 가장 효율성 측면에서 가장 활발하게 진행되고 있다. 이와 더불어 3D 레이저스캐닝 기술을 통해 3차원 공간정보를 획득하여 지형 변환량을 산정하는 연구도 다양하게 수행되고 있음.

본 연구에서는 데이터 측정의 경제성과 효율성을 고려하여 드론과 라이다를 활용하여 지형 변환량을 계산하였으며 효율적인 위험 예측을 위한 메타 데이터 관리 체계와 정보 전파 체계를 구현하였음.

다음의 그림은 전체 시스템의 구조이다.



지형 변환량 분석을 위해서는 지형 측정을 위한 클라이언트 시스템(GFMS/DART)와 데이터 저장, 분석, 전파를 위한 서버 시스템(GFMS/MART)으로 구성된다. 각 시스템의 구성요소들은 다음과 같다.

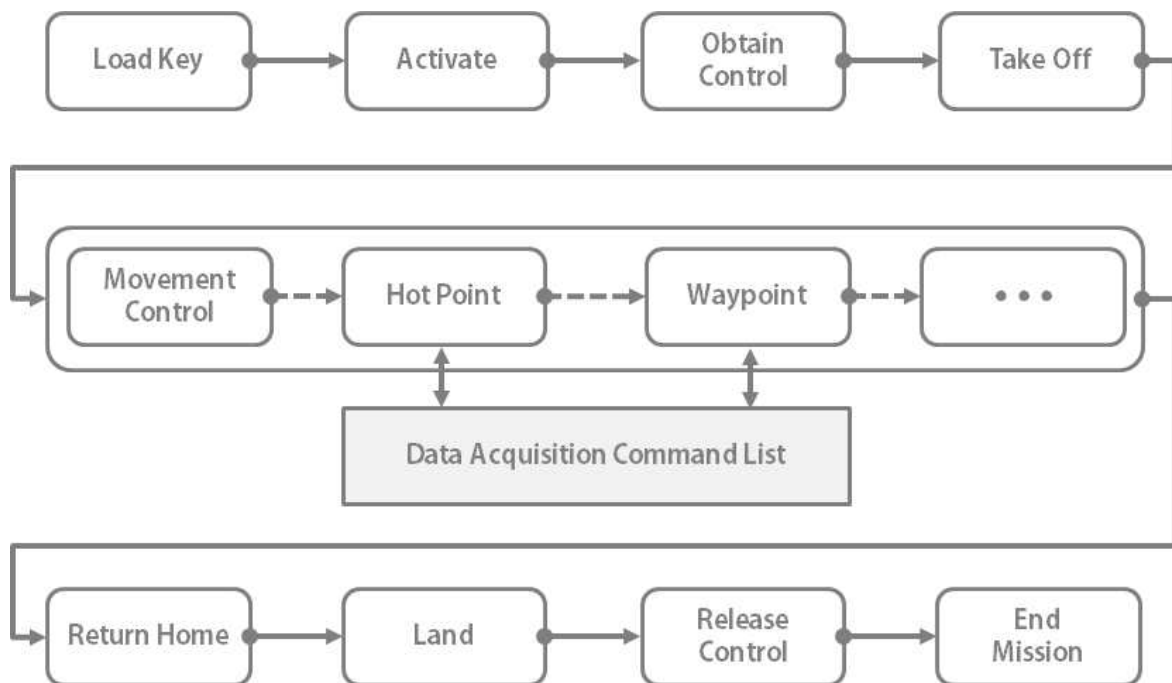
(2) 클라이언트 시스템(GFMS/DART)

클라이언트 시스템의 개념적인 구조는 다음과 같다.

클라이언트 시스템(GFMS/DART)		
응용 단계	Work Plan Loader	Auto Flight Module
	LiDAR Control Module	Data Communication Controller
드론 파트	Flight Controller	GPS
	IMU/Gimbal	Independent Communication Controller
라이다 파트	LiDAR Body	Communication Controller
로컬 스토리지	내부 저장장치	SD Card

① 작업 계획(Work Plan)의 적재 모듈

작업 계획 적재 모듈은 이동점 리스트, 획득장치, 스트리밍 여부, 부가 속성들을 포함한 정보들을 클라이언트에 적재하여 작업 수행하는 데이터로 사용한다. 작업 계획을 통해 클라이언트 프로그램이 실행되는 프로세스의 흐름도는 다음과 같다.



클라이언트에서 전달되는 드론과 라이다에 전달되는 메시지의 구조는 XML(eXtensible Markup Language)의 구조를 사용한다. 이와 같은 구조는 MAVLink 프로토콜과 호환 될 수 있도록 적용된다. 작업 계획의 개념적인 자료구조는 다음과 같음.

구분	데이터 형식	개요
Constructor	method	생성자
Coordinate	property	좌표값
Altitude	property	고도
Heading	property	방향
Corner Radius	property	코너각
Turn Mode	property	방향전환 모드
Gimbal Pitch	property	짐벌 피치
Speed	property	속도
Shoot Photo Time Interval	property	촬영 시간 인터벌
Shoot Photo Distance Interval	property	촬영 거리 인터벌
Action Message	property	명령어 메시지

- 작업 계획의 MAVLink 프로토콜 변환

MAVLink는 프로토콜은 통신을 통해 드론 비행체를 제어할 수 있는 프로토콜로 대부분의 드론 제조사들이 채택하고 있는 헤더만 가지는 메시지 마샬링(marshalling) 라이브러리이다. 컴퓨터와 연결된 드론은 자동 비행과 작업을 위해 해당 프로토콜을 사용하여 드론을 제어하게 된다. MAVLink는 모던 하이브리드 publish-subscribe 구조와 point-to-point 디자인 패턴을 따른다. MAVLink는 매우 제한된 통신 대역을 가지는 어플리케이션에 최적화되어 있어서 추가적인 프레임이 필요없다. 프로그래밍 방식도 제한된 메모리와 플래쉬 메모리의 리소스 제약이 있는 시스템에 최적화되어 있다. 많은 제조사가 다른 컴포넌트들끼리 상호운용성이 가능한 인터페이스를 제공하는 다양한 제품이 이미 출시되어 가장 널리 사용되고 있다.

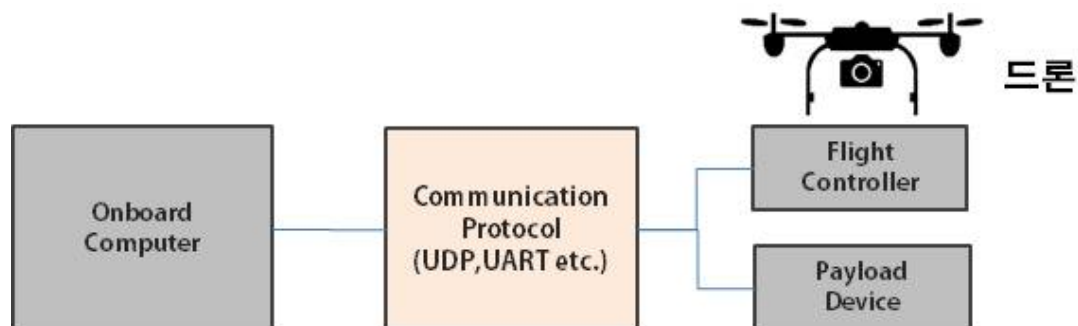
MAVLink는 2009년 로렌츠 마이어(Lorenz Meier)에 공개되었고 GPL(General Public License)로 활용되고 있음. 데이터 전송 단위인 패킷(packet)의 구조는 다음과 같이 Magic Packet을 포함해서 최소 8개에서 최대 263 바이트의 데이터를 포함한다.

MAVLink Frame : 8~263bytes



구분	바이트 색인	설명
STX	0	패킷의 시작을 알리는 바이트
LEN	1	페이로드(메시지)의 길이
SEQ	2	패킷 순서
SYS	3	시스템 아이디
COMP	4	컴포넌트 아이디
MSG	5	메시지 아이디
PAYLOAD	6~N+6	메시지
CKA,CKB	N+7,8	체크섬 바이트

- 작업 계획의 드론 전송 : 실제 드론의 비행 제어장치와의 통신은 제조사마다 약간의 차이가 있지만 제공되는 네트워크 프로토콜이 상이할 수 있다. 이 프로토콜은 구현 방법에 따라 다양한 형태가 있으며, 인터넷의 경우 대표적으로 TCP와 UDP 프로토콜을 제공하며 직접 연결을 위해서는 SPI, UART, CAN, I2C 통신 방식 등을 활용한다.



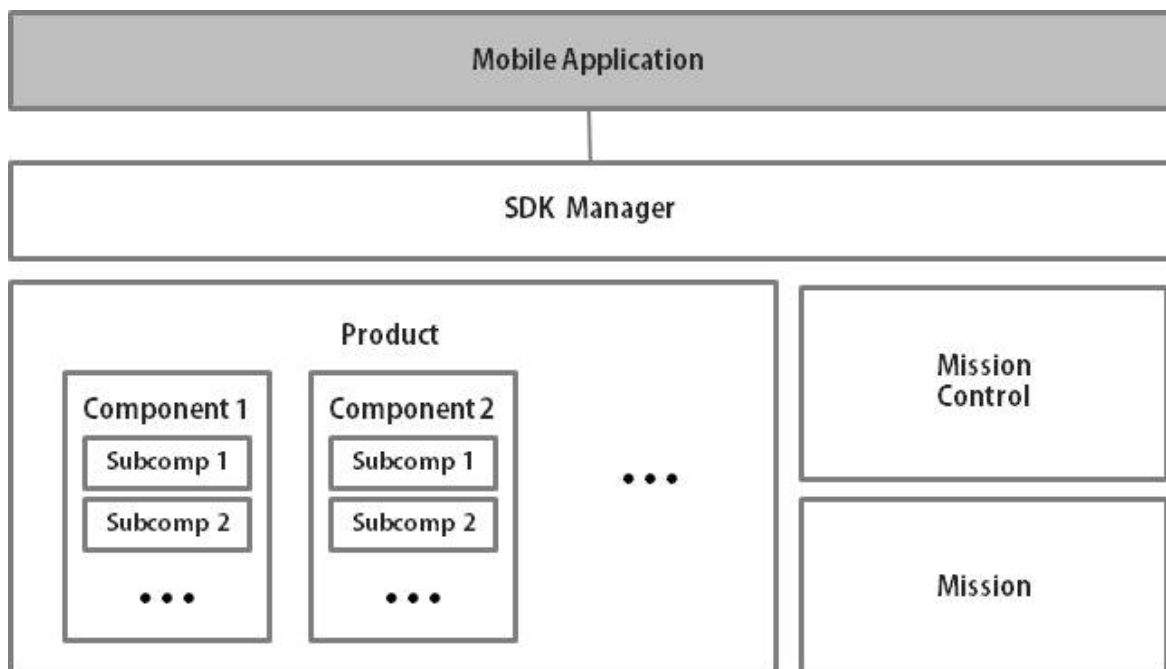
- 제조사 종속적인 라이브러리의 이용 : 작업 계획의 수행을 위해서 드론 제조사에 종속적인 어플리케이션 라이브러리를 활용할 수 있다. 이는 통신 방식과 같은 물리적인 구조를 이해하지 않아도 함수 단위의 메소드를 프로그래밍 코드에 직접 사용할 수 있다는 장점을 가지고 있다. 대표적인 드론 제조사인 DJI의 경우 이 방식을 제공하고 있다. DJI SDK라고 불리는 개발 라이브러리를 서로 다른 컴퓨터 플랫폼에서 직접 사용이 가능하다. DJI 개발 도구들은 임베디드 환경 뿐만 아니라 윈도우즈, 리눅스, 안드로이드, iOS 운영체제에서 모두 사용이 가능하다.

개발도구를 이용하여 드론의 제어가 가능한 영역들은 다음과 같음.

구분	주요 기능
비행(Flight) 제어	- Flight Action - Advanced flight control - Aircraft state data - Obstacle avoidance
카메라(Camera) 제어	- Camera and gimbal control - Live video feed - Stereo vision video feed - Real-time disparity map
미션(Mission) 제어	- Custom local-frame navigation - Pre-defined missions
서브시스템(Subsystem) 제어	- Remote Controller - Mobile SDK communication - Multi-Function I/O ports - Synchronization with DJI flight controller

② 자동 비행(Auto Flight) 모듈

자동 비행 모듈은 드론 종속형 자동 비행 라이브러리 또는 MAVLink(Micro Air Vehicle Link) 프로토콜을 이용하여 드론의 비행을 제어하는 모듈이다. 자동 비행을 포함한 드론 연동 어플리케이션의 구조는 다음과 같다.



구분	주요 기능
SDK Manager	- SDK 등록 및 인증 - 제품 연결 - 자제 실행
Product	- 드론 비행체 관련 메인 클래스 - 메인 컴포넌트의 시작점
Component	- 컴포넌트 클래스 - 짐벌, 카메라, 비행 제어기, 리모콘 - 컴포넌트 제어 및 상태 정보 확인 - 서브컴포넌트 제어
Mission	- 독립적인 미션 단위 설정 클래스 - Waypoint, ActiveTrack 등의 미션 - 단위 미션의 속성 설정
Mission Control	- 단위 미션의 수행을 제어 - 독립 미션의 수행 - 타임라인 기반 다중 미션 수행

자동 비행 모듈에서 사용된 테스트 비행의 내용은 다음과 같다.

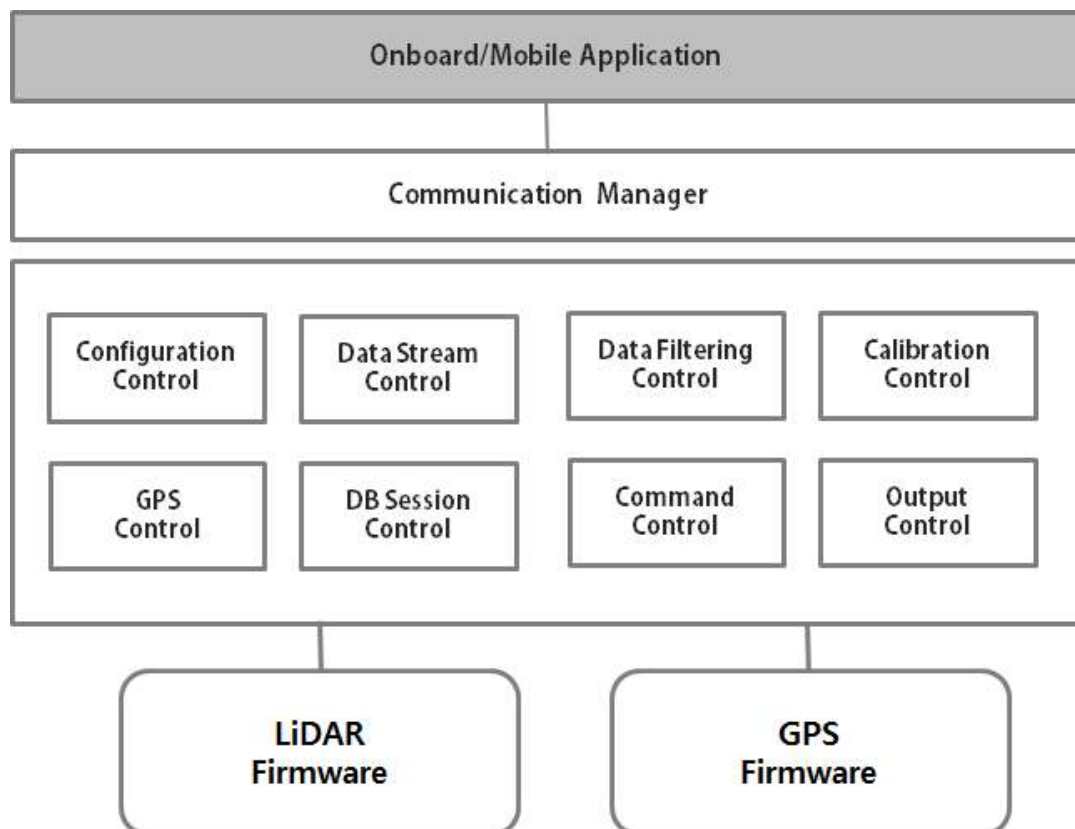
연번	구분	위도	경도	고도	비고
1	시작점	37.403082	126.931405	90	
2	이동점-1	37.403082	126.931405	120	
3	이동점-2	37.402834	126.930598	120	
4	이동점-3	37.402701	126.931107	120	
5	도착점	37.403082	126.931405	90	



③ 라이다 데이터 입력 모듈

라이다 시스템(LiDAR, Light Detection and Range)은 관측 대상물의 표면에 조밀한 간격으로 무수히 많은 레이저 펄스를 주사하여 얻은 X,Y,Z형태의 3차원 정보를 고밀도로 직접 취득할 수 있으며, 그 처리 과정 또한 자동화가 가능하여 현황 측량의 대안으로 많이 활용되고 있다.

라이다 데이터 입력 모듈은 드론의 온보드 컴퓨터에서 실시간으로 라이더 데이터를 입력 받아 서버에 로깅하는 모듈이다. 라이다 데이터와 드론을 연동 어플리케이션의 구조는 다음과 같다.



구분	주요 기능
Communication Manager	- Device File 개방 - 제품 연결 - 상태 정보 획득 - 통신 설정값 지정 - 통신 방식 설정
Configuration Control	- 라이다 포트 및 IP 정보 - 위치 및 오리엔테이션 설정 - Reflector 방식 설정 - 반환 방식 설정 - 회전 속도 설정 - Garbage data 설정 - 데이터 구분자(바이너리 데이터 템플릿) 설정
Data Stream Control	- 스트리밍/파일 모드 설정 - 장치 파일 지정자 개방 - 데이터 입/출력 - 분석용 데이터 적재
Data Filtering Control	- 항목별 절대 필터링 값 설정 - 필터링 방식별 필터링 수식 설정 - 필터링 항목 설정 - 실시간/배치 필터링 설정
Calibration Control	- Calibration 파일 변환 - GPS 파라미터 변환 - 네트워크 파라미터 변환
GPS Control	- GPS 환경변수 설정 - 데이터 입/출력 - 라이더 데이터 동기화
DB Seession Control	- DB 연결 및 인증 설정 - DB 데이터 로깅 - 데이터 동기화
Command Control	- 장치 초기화 - 장치 설정 - 명령어 전송
Output Control	- 데이터 형식 지정 - 데이터 로깅 - 데이터 변환

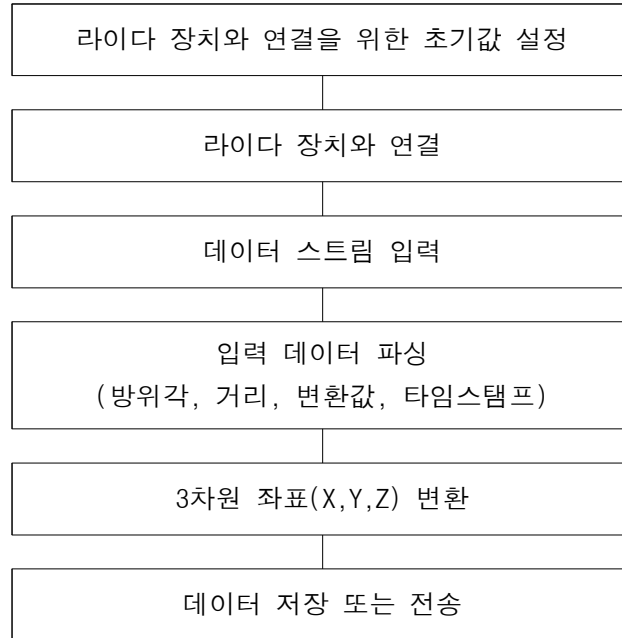
- 실험용 LiDAR 규격

본 연구에서는 실험용 라이다로 벨로다인사의 VLP-16 제품과 스캔스사의 Sweep 1.0을 사용하였으며 그 중 정밀도와 측정 범위가 우수한 VLP-16 제품의 규격과 특성은 다음과 같다. 특히 VLP-16은 측정 대상의 특성에 따라 Reflector 방식을 지정함으로써 정밀도를 향상 시킬 수 있다.

구분	설명
제품명	Velodyne VLP-16
Laser/Detector 개수	16개
수평 회전각	360°
수직 회전각	30°
회전 속도	5~20 Hz
측정범위	100미터
측정오차	+/- 3cm
수평각 해상도	0.1°~0.4°
수직각 해상도	2°
가동고도	200m
장치 I.P	192.168.1.201
호스트 I.P	196.168.1.77
해상도	256 비트
레이저 타입	Class 1
레이저 규격	903 nm
소비전력	8W
제품사진	

- LiDAR 데이터 처리 프로세스

LiDAR 데이터의 직접 입력을 위한 양방향 통신을 위해서는 해당 제품의 프로토콜을 분석하여 3차원 좌표로 실시간 또는 저장 방식으로 처리하여 한다. 다음은 라이다 데이터 처리되는 개념적인 흐름도를 보여준다.



라이다 데이터로부터 3차원 좌표를 변환하기 위해서는 우선 입력 데이터를 파싱해야 하며 결과적으로 수직각(ω), 수평각(α), 거리(r)를 추출할 수 있다. 수직각은 고정값으로 레이저 ID를 통해 확인할 수 있다. 다음의 공식은 3차원 좌표인 X , Y , Z 를 추출한다.

$$\begin{aligned}
 X &= r \cdot \cos(\omega) \cdot \sin(\alpha) \\
 Y &= r \cdot \cos(\omega) \cdot \cos(\alpha) \\
 Z &= r \cdot \sin(\omega)
 \end{aligned}$$

- LiDAR 데이터 구조

LiDAR 데이터는 UDP Ethernet 패킷의 구조로 호스트 컴퓨터로 전송된다. 패킷은 라이다로부터 전송되는 데이터(data) 패킷과 GPS로부터 전송되는 위치(position) 패킷으로 구분된다. 데이터 패킷의 크기는 전체 1248바이트이며 거리값, 보정변환값, 방위각, 타임스탬프, 제조사코드, 반환방식 등을 포함한다. UDP 포트 2368번을 통해 송수신할 수 있다.

헤더	데이터 블록-1	데이터 블록-1	...	데이터 블록-12	타임스탬프	제조사코드
42바이트	100바이트	100바이트		100바이트	4바이트	2바이트

- 방위각(azimuth) 환산

- 2 바이트 방위각 바이너리 데이터 값 적재 : 예) 0x33, 0x71
- 바이트 데이터 역순화 : 예) 0x71, 0x33
- 역순화 바이트를 비부호 2 바이트 정수로 결합 : 예) 0x7133
- 10진수로 변환 : 예) 28,979
- 100으로 나눔 : 예) 289.79
- 결과값 : 289,79o

- 거리값(distance) 환산

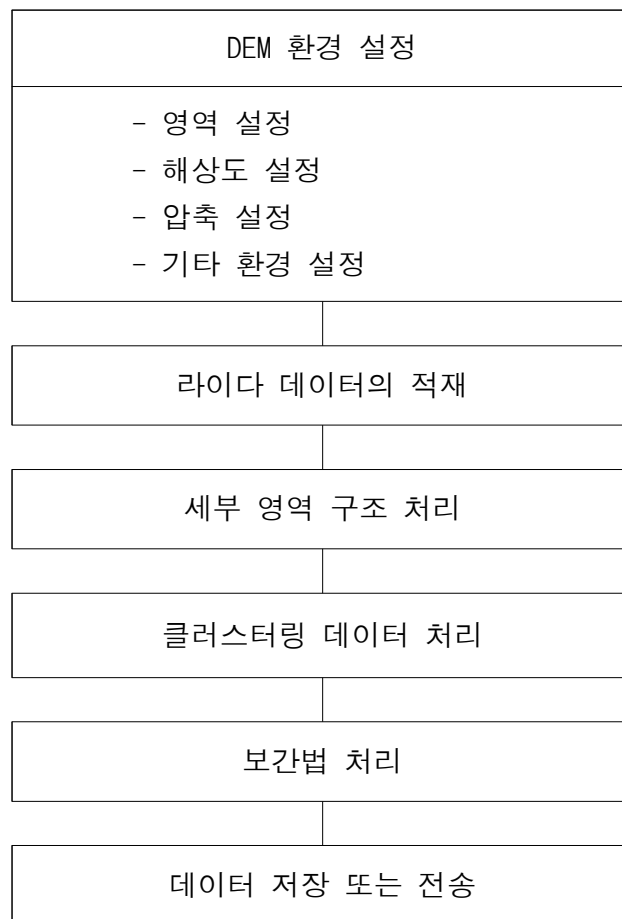
- 2 바이트 방위각 거리 데이터 값 적재 : 예) 0x89, 0x59
- 바이트 데이터 역순화 : 예) 0x59, 0x89
- 역순화 바이트를 비부호 2 바이트 정수로 결합 : 예) 0x5989
- 10진수로 변환 : 예) 22,921
- 2mm를 곱함 : 예) 45,842mm
- 결과값 : 45,842 미터

- 타임스탬프(timestamp) 환산

- 4 바이트 타임스탬프 데이터 값 적재 : 예) 0x61, 0x67, 0x89, 0x5A
- 바이트 데이터 역순화 : 예) 0x5A, 0x89, 0x67, 0x61
- 역순화 바이트를 비부호 4 바이트 정수로 결합 : 예) 0x5A896761
- 10진수로 변환 : 예) 1,522,100,065 마이크로 초
- 1,000,000으로 나눔(초로 변환) : 예) 1,522.1000065 초
- 결과값(분,초로 변환) : 25분 22.1000065초

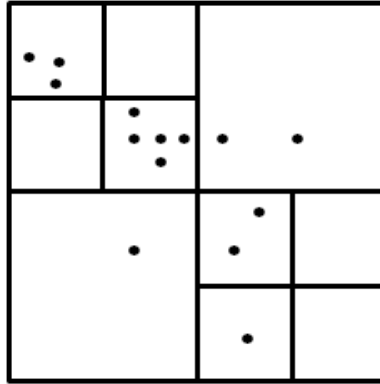
(4) 라이다 데이터 격자화

특정한 대상영역에서의 정확한 지형 변환량을 계산하기 위해서는 계산에 활용되는 모든 원시 라이다 데이터를 격자화하고, 그 후 각 격자에 대해 중첩된 다양한 정보에 적합한 변환량 계산 공식을 적용하여야 한다. 일차적으로 필터링된 LiDAR 데이터는 관리 표준을 위해 격자화된 DEM(Digital Elevation Model)으로 변환되어 저장된다. 실제 수많은 양의 점군(point set) 데이터를 변환하려면 프로세서에 부하가 많이 걸리므로 병렬적으로 수행하는 방식이 유리하다. 다음은 라이다 데이터가 격자화되는 개념적인 흐름도를 보여준다.



- 세부 영역 구조 처리

쿼드 트리를 이용한 격자 구성에 점군을 포함시키기 위한 사각 구조(Bounding box)를 평면에 지정한다. 쿼드 트리의 단말 노드에는 범위에 포함하는 점군들을 배치하게 된다. 트리의 레벨과 균형을 고려하여 병렬 처리 할 수 있게 구조화하며 각 영역들은 다시 4분면으로 순환적으로 추가된다. 다음의 그림은 2단계 쿼드 트리의 개념적인 구조를 보여준다.



- 클러스터링 데이터 처리

쿼드 트리의 영역의 리프 노드에 점군을 분류하여 매핑하는 프로세스로 점들의 값을 통하여 순환적으로 트리를 순회하여 리프 노드에 연결된다.

- 보간법 처리

전체 쿼드 트리 영역의 리프 노드의 값을 보간하여 전체 영역에 대한 격자를 완성한다. 필터링이 필요한 경우 데이터 값에 대한 필터링을 부가적으로 수행한다.

(5) 격자 데이터를 통한 지형 변환량의 산출

격자화된 DEM을 통해 변환량을 계산하기 위해서는 대상단면의 체적(A)과 단면 사이의 리(l)를 이용하는 적절한 변환량 계산공식을 적용하여야한다. 변환량을 구하고자 하는 대상지역의 단면의 면적을 A_1, A_2 라 하고 그 간격을 l 이라 하면, 최종적인 변환량 V 는 다음과 같이 표현된다.

$$V = \frac{1}{2}(A_1 + A_2) \times l \quad (2)$$

중앙단면법에서는 대상지역의 중앙단면을 A_m 이라하고 양 단면간의 거리를 l 이라하면, 변환량 V 는 다음과 같이 표현된다.

$$V = A_m \times l \quad (3)$$

각주공식에서는 대상지역을 다각형으로 표현하는데, 이때 다각형의 위·아래의 면이 평행이고 측면도 모두 평면으로 된 다각형의 부피, 즉 변환량 V 는 다음과 같이 표현된다.

$$V = \frac{h}{6}(A_1 + A_m + A_2) \quad (4)$$

대상지역의 최종적인 변환량은 실제 지형에 대해 상기의 변환량 계산공식을 적용하여 계산 되는데, 여기서는 수치지형자료를 처리하여 격자형 데이터를 구성하였으므로 각 격자에서의 DEM의 표고값과 계획평면도의 표고값을 비교하여 체적을 산정한 후 격자의 크기를 적용하여 DEM의 각 격자에서의 변환량을 산정하는 방안을 적용한다.

본 연구에서는 격자의 각 교차점에서의 표고값을 추출하고 수치표고모델과 계획 표고데이터의 표고값의 차이를 계산하여 사다리꼴 면적 계산공식에 대입하여 각 격자 단면에서의 면적을 계산하였다. 격자의 교차점에서의 표고값은 표고값의 변화는 선형이라고 가정한 후 다음의 그림 2과 같이 교차점에 인접하는 격자의 표고값의 평균으로 계산한다.

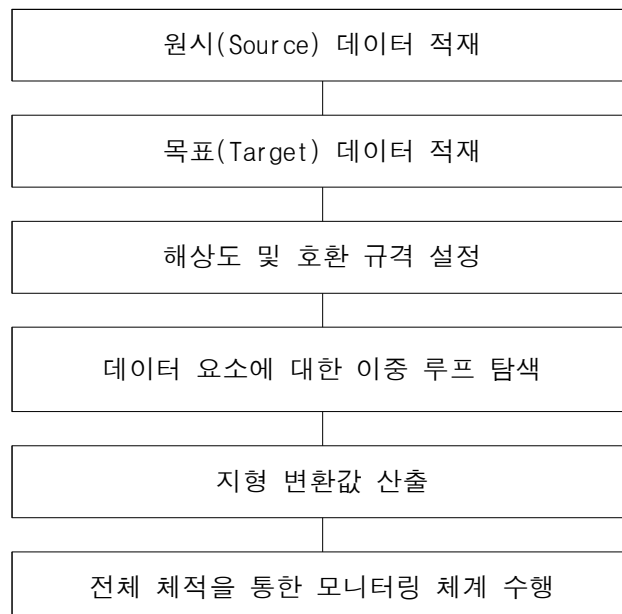
다음의 식 (5)는 격자 교차점에서의 일반적인 표고값 계산과정을 표현한 것이다.

$$h_n = \frac{1}{n}(v_1 + \dots + v_n) \quad (5)$$

여기서, h 는 대상 격자 교차점 표고값, v 는 인접격자 표고값, n 은 인접 격자의 수를 각각 나타낸다. 수치표고모델과 계획 표고데이터를 구성하는 격자 교차점에서의 표고값을 모두 산출한 후 각 교차점에서의 수치표고모델 격자 교차점 표고값과 계획표고데이터 격자 교차점 표고값의 차이를 산출하여 사다리꼴 체적 계산공식에 대입하여 면적을 산출한다. 다음의 그림 3은 격자 단면체적 계산과정을 나타내며, 식 6은 그림 3의 격자단면을 통해 적용되는 체적 계산공식을 나타낸 것이다.

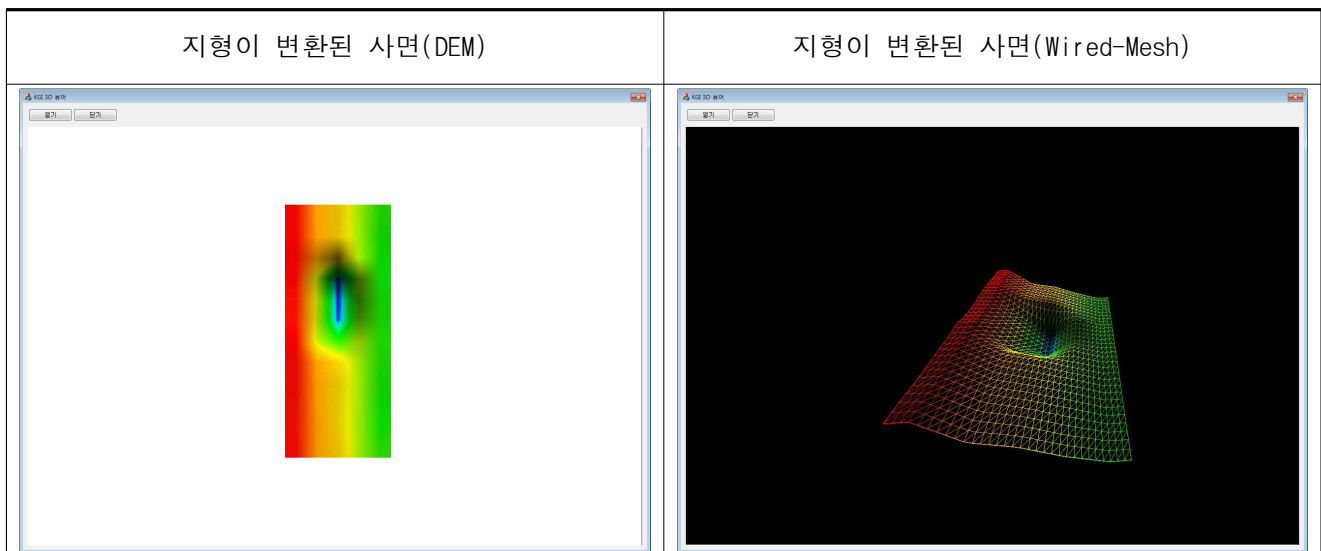
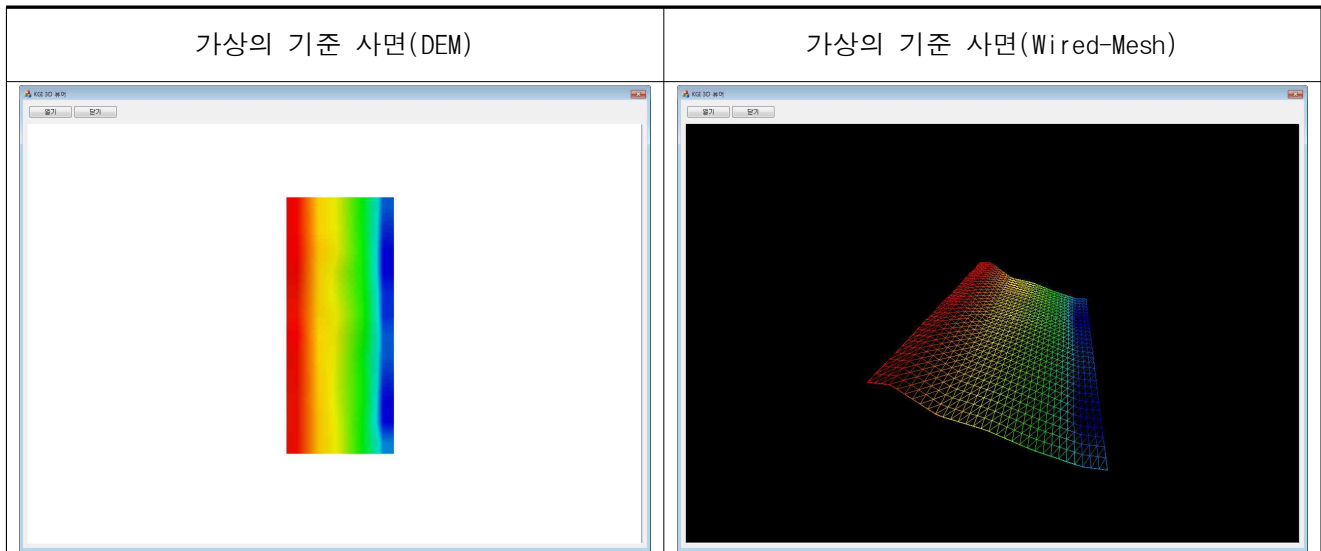
$$V = \frac{1}{2} \{ (Dh_1 - Ph_1) + (Dh_2 - Ph_2) \} \times l \quad (6)$$

변환량을 체적으로 산출하기 위한 프로세스의 흐름도는 다음과 같다.



(6) 지형 변환과 관련된 3차원 출력과 변환량의 산출

사면의 붕괴와 같은 지형 변환을 모니터링하기 위한 인터페이스로 격자 출력 및 와이어 메시 방식으로 구현하였으며 기준이 되는 사면과 변형된 사면의 매트릭스 연산을 통해 변형된 체적을 산출할 수 있다.



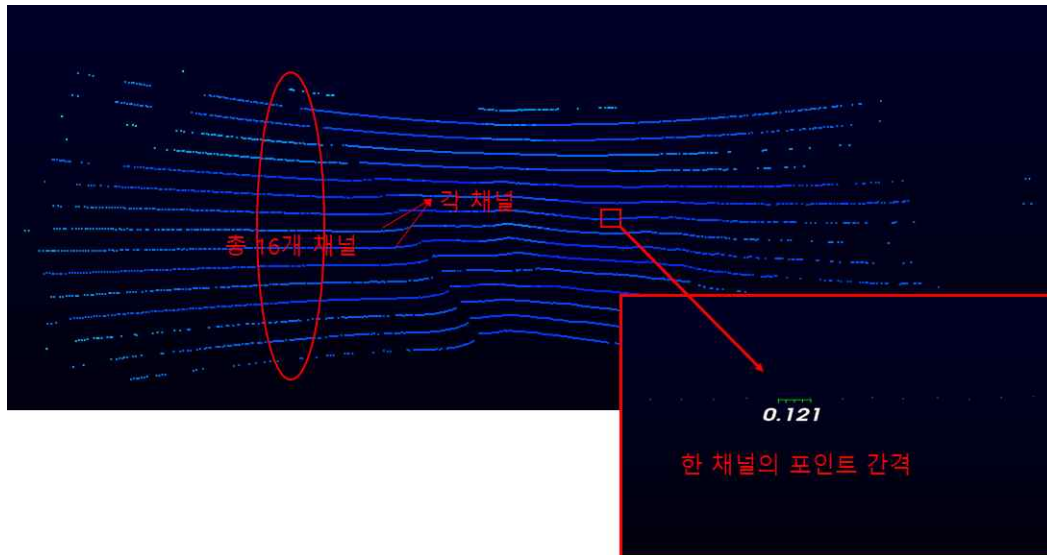
(7) 포인트 클라우드 개수 증대에 따른 처리용량 개선안

라이다의 현재 설정 값인 16채널, 600RPM일 때 한 프레임에 손실 없이 찍혀야 할 총 포인트 개수 계산은 한 채널을 기준으로,

- 포인트 간격 : 약 12cm
- 라이다 높이 : 3,000cm
- 라이다 수평범각 FOV(Field Of View): 90도
- 360도의 원을 생각했을 때 원의 둘레는 $2 \times 3.14 \times 3,000 = 18,840 \text{ cm}$
- 포인트 간격이 12cm 이므로 $18,840 / 12 = 1570$ 포인트
- 다시 90도는 360의 4분의 1이므로 $1570 / 4 = 392.5$
- 다시 16개 채널을 적용하면 $392.5 \times 16 = 6280$ 포인트

즉 한 프레임에 찍혀야할 포인트는 손실이 없을 때 약 6280포인트인데 연구과제 수행 시 측량된 데이터는 한 프레임 당 약 5500 ~ 6200 포인트이고 평균 5,850개 일 때 약 430개의 포인트 손실 발생하게 된다.

측량 대상물이 길이 400m, 폭 20m, 높이 15m일 때 총 프레임수가 평균 약 10,000개 정도이며 해당 측량 대상물에 대해 총 4,300,000개의 포인트는 손실 발생한다.



[라이다의 각 포인트]

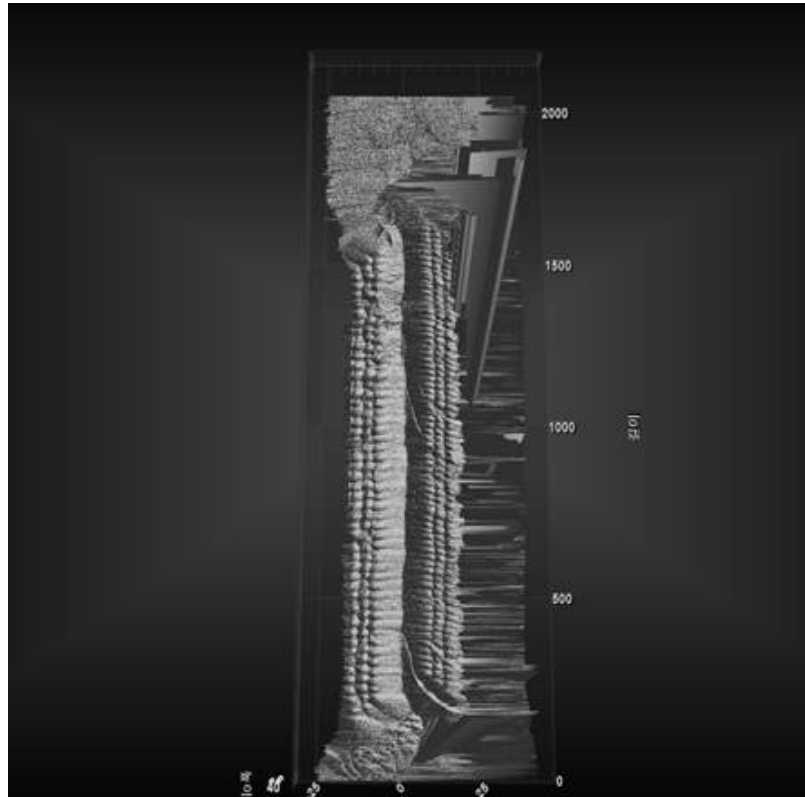
포인트의 손실 요인은 측정대상물의 표면질과 모양, 드론의 흔들림으로 인한 레이저 FOV의 범위 이탈 등으로 예상되며, 또한 드론의 흔들림은 고도의 변화로 인해 반사거리의 오차가 발생하여 측량계산에 영향을 주게 된다.

같은 면적에 대해 레이저 포인트 간격을 12cm에서 6cm로 더 세밀하게 방사하여 반사되는 포인트의 개수를 늘리고 짐벌의 장착으로 안정된 드론 비행을 도모함으로써 포인트 손실과 흔들림을 방지하여 측정물의 측량값의 오차를 최소화 할 수 있다. 또한 늘어난 포인트 클라우드의 처리 알고리즘은 라이다의 고도, 반사된 레이저의 길이, 채널과 채널 사이의 길이 등을 평균적으로 상수화하여 각 포인트마다 매번 계산하지 않고 적용함으로써 최대 전체 포인트 개수에 대해 계산하는 시간만큼을 줄 일 수 있다. 이때 발생하는 오차는 위에 서술한 라이다의 성능 개선으로 상쇄됨이 예상된다.

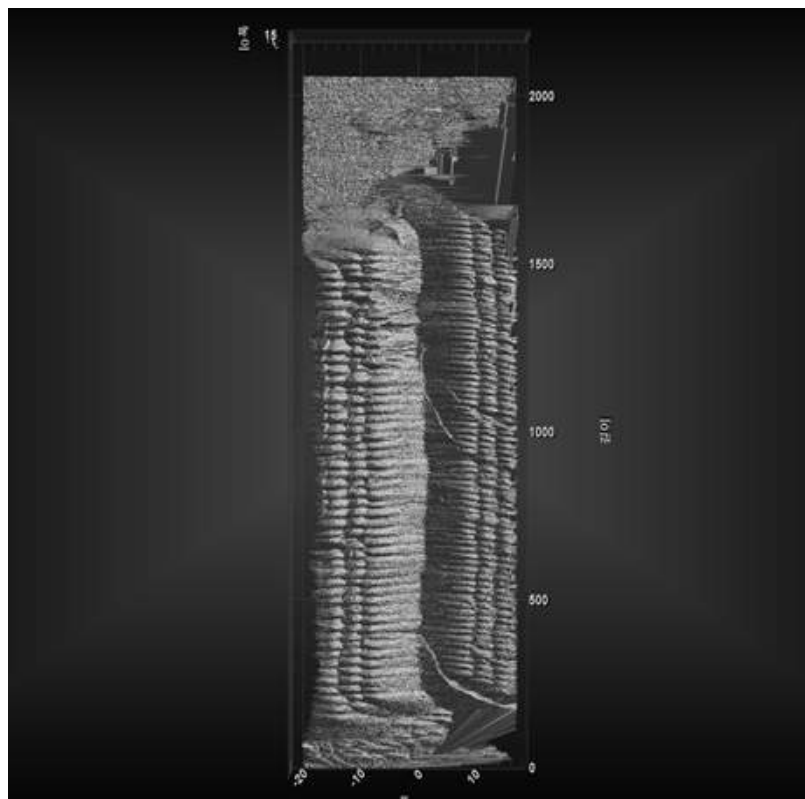
(8) 적용 사례

- 평지 자동 검출 알고리즘 적용으로 오차 개선
 - 드론에 설정된 일률적인 고도값을 사용함으로써 발생하는 측량 오차 제거
 - 측정된 파일의 첫 프레임에서 드론의 가장 높은 상위 10개의 값을 자동으로 구하여 그 평균값을 산출
 - 파일의 첫 프레임에서 평지 데이터를 산출하기 어려울 경우 마지막 프레임에서 드론의 평균 고도값을 산출 할 수 옵션 추가
- 노이즈 데이터 제거 기능 추가
 - 펜스, 크레인, 레일 등 고도 데이터 제거 기능 추가
 - 3D 화면을 통하여 제거해야할 범위의 거리 및 고도를 입력
 - 거리는 파일의 전면의 중심으로부터 왼쪽 및 오른쪽의 거리 값을 입력

- 평지를 바닥으로 하여 일정 높이 이상의 데이터를 제거하는 높이 값을 입력
 - 각 거리 및 높이 값을 같음하기 위해 화면에 표출된 3D 파일 이미지에 대해 거리 및 높이의 값이 미터로 표시
- 드론과 라이다의 성능 및 시스템 개선을 통한 측청 결과 비교



[개선 전 측청 결과]



[개선 후 측청 결과]

- 드론의 라이다 센서에 설치된 진동 방지 시스템을 탑재하고 개선된 측청 시스템 적용 후 위의 그림처럼 가시적으로도 고도화 전보다 잘 측청되었음을 알 수 있다.

4. 무선(블루투스 , WiFi/LTE) 네트워크를 이용하여 계측기와 연동이 가능한 스마트폰용 통신 관리 앱 모듈 개발

(1) 블루투스 무선통신 비콘과 연동된 스마트폰 앱 구현

○ Midascon 비콘 연동 앱 개발

- Midascon 비콘 SDK를 이용하여 비콘과 연동되는 안드로이드용 앱을 개발하였다. 개발툴은 구글사에서 제공하는 안드로이드 스튜디오를 사용하였으며 비콘 SDK는 안드로이드의 개발 언어인 자바를 사용하였으나 본 과제에서는 최근 자바보다 더 간편하고 개발속도가 빠른 코틀린을 사용하였다. 자바와 코틀린의 대표적인 차이는 다음과 같다.

• 상속

//java

```
public class MainActivity extends AppCompatActivity { ... }
```

//kotlin

```
class MainActivityKt : AppCompatActivity, OnClickListener { ... }
```

kotlin에서는 상속의 개념을

extends 대신 *콜론 :* 으로 표현

인터페이스 추가는 , 로 추가

• findViewById

//java

```
private Toolbar toolbar;
```

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState){
```

```
    super.onCreate(savedInstanceState);
```

```
    setContentView(R.layout.activity_main);
```

```
    Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
```

```
... }
```

//kotlin

```
private val toolbar by lazy {
```

```
    findViewById(R.id.toolbar)
```

```
} lazy 라는 키워드를 통해서 findViewById
```

lazy는 onCreate가 완료된 후 findViewById 하므로 null이 안뜸

- **OnCreate**

```
//java
@Override
protected void onCreate(@Nullable Bundle savedInstanceState) { ~ }
```

//kotlin

```
override fun onCreate(SavedInstanceState: Bundle? ) { ~ }
```

java : null 이 허용이 되어 null 처리에 대한 부분이 따로 존재 하지 않음

kotlin : ? 가 있으면 이 변수는 무조건 null이 될 수 있음

android 에서 @Nullable 과 같은 annotation

- **lambda 식**

```
//java
fab.setOnClickListener ( new View.OnClickListener () {
    @Override
    public void onClick(View view) {
        ...
    }
});
```

//kotlin

```
fab.setOnClickListener {
    view ->
    ...
}
```

```
fab.setOnClickListener {
    ...
}
```

fab : 플로팅액션버튼

OnClickListener 혹은 onClick(View view) 등 생략가능

코틀린에는 세미콜론 ; 이 붙지 않아 엔터로 대신함

- Midascon SDK 연동하여 블루투스로 계측치를 수신 받기 위해서는 안드로이드 앱의 퍼미션을 설정하여야 한다. 개발 프로젝트의 AndroidManifest.xml에 다음과 같이 설정한다.

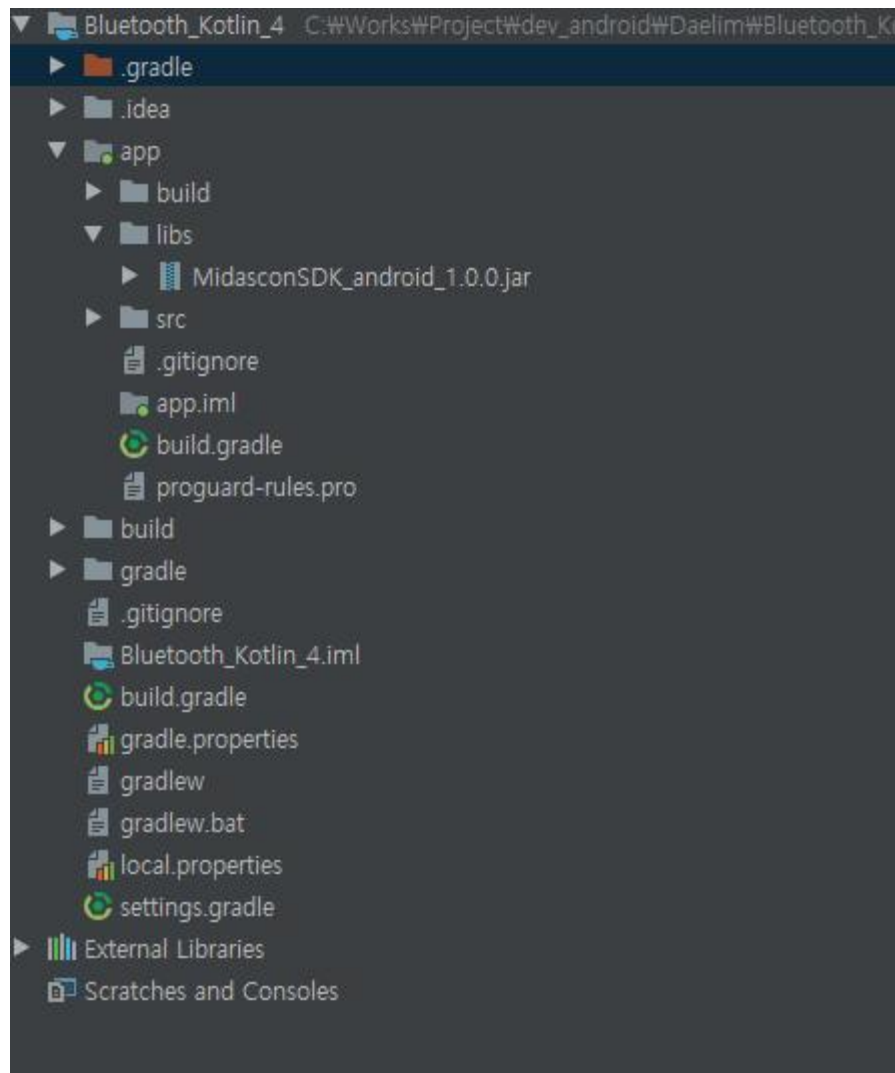
```

<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />

```

[블루투스 퍼미션 설정]

- Midascon SDK 가 제공하는 라이브러리인 “MidasconSDK_android_1.0.0.jar” 을 임포트한다. 개발 전 준비가 완료되면 아래와 같은 프로젝트 구성이 된다.



[비콘SDK를 이용한 프로젝트 구성]

- 비콘콜백 등록

- BLE 스캔으로 발견되는 비콘 데이터를 전달 받을 콜백을 등록한다.
- 콜백 등록 com.hanvitsi.midascon.BeaconCallback 인터페이스를 등록한다.

```
// ContextManager 호출
contextManager = getMidasApplication().getContextManager();

// 모든비콘 수신, true일 경우 마이다스콘 프로토콜만 수신
contextManager.getBeaconSettings().setMidasScanMode(false);

// 비콘 콜백 등록
contextManager.setBeaconCallback(new BeaconCallback() {

    @Override
    public void onBeaconCallback(int code, Beacon beacon) {

    }

});

//스캔 시작
contextManager.startLeScan();
```

- 응급신호콜백 등록

- 마이다스콘의 응급신호를 를 전달 받을 콜백을 등록한다.
- 콜백 등록 com.hanvitsi.midascon.EmergencyCallback 인터페이스를 등록한다.

```
// ContextManager 호출
contextManager = getMidasApplication().getContextManager();

// 모든비콘 수신, true일 경우 마이다스콘 프로토콜만 수신
contextManager.getBeaconSettings().setMidasScanMode(false);

// 응급 콜백 등록
contextManager.setEmergencyCallback(new EmergencyCallback() {

    @Override
    public void onEmergencyCallback(int code, Beacon beacon) {
        // TODO Auto-generated method stub
    }

});

// 스캔 시작
contextManager.startLeScan();
```

- 안드로이드 기기와 블루투스 기기간의 연동은 BluetoothAdapter라는 클래스를 통해서 직접 구현해야 하지만 비콘 SDK는 내부적으로 이 클래스를 통하여 비콘기기의 고유의 UUID로 안드로이드기기와 연동된다.

```

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)

    contextManager = midasApplication.contextManager // 블루투스 연동
    contextManager!!.beaconSettings.isMidasScanMode = true // 비콘의 블루투스만 연동

    textView_1.text = "수신대기중"

    val pref: SharedPreferences = getSharedPreferences("비콘무선계측", Activity.MODE_PRIVATE)
    editText_ServerIP.setText(pref.getString("ServerIP", ""))
    editText_ServerPort.setText(pref.getString("ServerPort", "80"))
}

```

- 비콘과 연동 후 계측치는 어느 시점에서 수신 될지 모르므로 콜백함수를 구현하여 등록하여야 한다. 또한 항상 수신을 대기하고 있어야 하므로 스레드로 실행을 해야 한다.

```

override fun onBeaconCallback(status: Int, beacon: Beacon) { // 콜백함수 정의
    when (status) {
        BeaconCallback.STATUS_CODE_ENTER, BeaconCallback.STATUS_CODE_UPDATE -> {
            textView_1.text = "주소: " + beacon.mac
            textView_2.text = "온도: " + beacon.temperature.toString() + " °C"
            textView_3.text = "습도: " + beacon.humidity.toString() + "%"

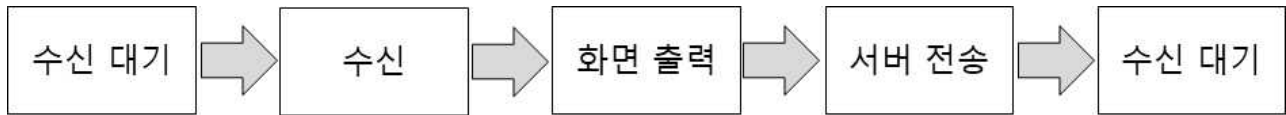
            HttpSendData(beacon.mac, beacon.temperature.toString(), beacon.humidity.toString())
        }

        BeaconCallback.STATUS_CODE_EXIT -> {
            textView_1.text = "수신대기중"
            textView_2.text = ""
            textView_3.text = ""
        }
        else -> {
            textView_1.setText(R.string.beacon_unkown_code)
        }
    }

    runOnUiThread(this) // 스레드로 실행
}
override fun run() {
    if (adapter != null)
        adapter!!.notifyDataSetChanged()
}

```

- 블루투스 비콘으로부터 계측치를 수신 받아 원격 서버에 데이터를 송신하는 과정은 다음과 같다.



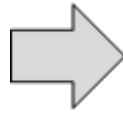
무선 비콘 계측

IP : 192.168.25.45

포트 : 80

수신 대기 중

수신 대기



무선 비콘 계측

IP : 192.168.25.45

포트 : 80

주소 : 74-f0-7d-ce-4f-c4

온도 : 24.87 °C

습도 : 31.9 %

수신, 출력, 전송

무선 비콘 계측

IP : 192.168.25.45

포트 : 80

주소 : 74-f0-7d-ce-49-8a

온도 : 24.75 °C

습도 : 31.56 %

서버와 연결이 되지 않습니다.

무선 비콘 계측

IP : 192.168.25.45

포트 : 80

주소 : 74-f0-7d-ce-49-8a

온도 : 24.7 °C

습도 : 31.47 %

서버로부터 응답이 없습니다.

무선 비콘 계측

IP : 192.168.25.45

포트 : 80

주소 : 74-f0-7d-ce-4f-c4

온도 : 24.96 °C

습도 : 32.2 %

서버와 연결이 끊겼습니다.

[데이터 전송 중 각종 오류]

- 안드로이드에서 서버로의 데이터 송신은 HTTP통신을 이용하였는데 안드로이드 통신 라이브러리 중 상용 서비스에서 많이 적용되어 있는 OkHttpClient를 사용하였다. OkHttpClient는 써드파티 라이브러리로 HTTP 기반의 request/response 방식을 사용한다. 다음과 같이 프로젝트에 추가하였다.

- 앱의 build.gradle에 라이브러리 포함

```
implementation 'com.squareup.okhttp3:okhttp:3.8.0'
```

- AndroidManifest.xml에 인터넷 퍼미션 추가

```
<uses-permission android:name="android.permission.INTERNET"/>
```

- 데이터는 HTTP Post 방식으로 전송하는데 HTTP의 body 부분에 데이터를 첨부하여 key값과 value 값을 매핑하여 작성 한다. HTTP의 request를 정의하고 마지막으로 Callback 메서드로 서버의 response에 대한 처리를 담당하는 코드를 작성하였다.

```
val pref: SharedPreferences = getSharedPreferences("비콘무선계측", Activity.MODE_PRIVATE)
val editor: SharedPreferences.Editor = pref.edit()
editor.putString("ServerIP", editText_ServerIP.text.toString())
editor.putString("ServerPort", editText_ServerPort.text.toString())
editor.commit()

textView_4.text = ""

var url:URL
// URL
try {
    url = URL(
        "http",
        editText_ServerIP.text.toString(),
        Integer.parseInt(editText_ServerPort.text.toString()),
        "beacon.html"
    )
} catch(e: Exception){

textView_4.text = "서버URL이잘못지정됐습니다."
return;
}

//데이터
val requestBody: RequestBody = FormBody.Builder()
    .add("mac",mac)
    .add("temperature",temperature)
    .add("humidity",humidity)
    .build()

// OkHttpClient Request
val request = Request.Builder()
    .url(url)
    .post(requestBody)
    .build()

// 클라이언트생성
val client = OkHttpClient()

// 요청전송
client.newCall(request).enqueue(object : Callback {

override fun onResponse(call: Call, response: Response) {

}

override fun onFailure(call: Call, e: IOException) {
    if(e.message == "timeout"){
        textView_4.text = "서버로부터응답이없습니다."
    }
}
```

```

else if(e.message!!.substring(0, ("failed to connect to").length) == "failed to connect to") {
textView_4.text = "서버와연결이되지않습니다."
}
else if(e.message!!.substring(0, ("unexpected end of stream on").length) == "unexpected end of
stream on") {
textView_4.text = "서버와연결이끊겼습니다."
}
else {
textView_4.text = e.message
}
call.cancel()
}
})

```

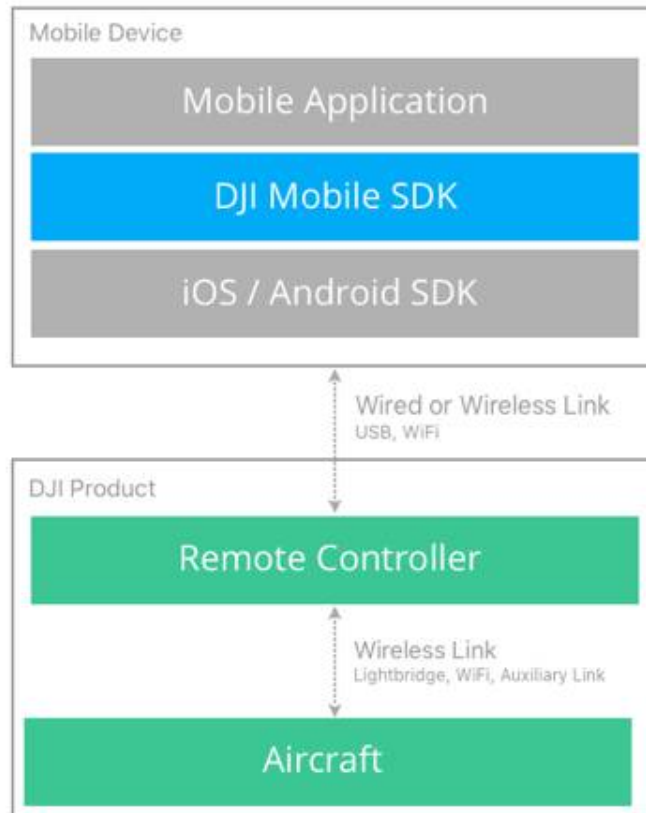
(2) 드론의 자율 비행을 위한 경로 설정 스마트폰 앱 구현

○ DJI SDK를 이용한 경로 설정 앱 개발

- 드론 제작사 인 DJI에서 제공한 안드로이드용 SDK의 경로설정API를 이용하여 개발 하였다. 안드로이드용 SDK는 DJI 사이트에서 내려 받을 수 있다. 앱을 개발하기 위해서는 안드로이드 스튜디오 3.0을 사용하고 탐색을 위해 Google Map API를 사용하였다.
- DJI SDK 모바일 앱의 경우 앱의 활성화를 위해 드론과 사용자의 DJI 계정을 바인딩 해야 한다. 앱이 활성화되어 있지 않거나 바인딩되지 않은 경우 또는 낮은 버전의 SDK를 사용 중인 경우는 일부 기능이 비활성되고 비행거리와 높이가 제한된다.
- 경로 설정을 위해 지도 뷰를 사용하여 경로를 표시하고 경로를 따라 비행을 실행할 때 드론의 비행 경로를 표시 할 수 있다. 본 과제에서는 Google Map으로 지도 뷰를 제작 하였다.

○ 모바일 DJI SDK 의 구성

- 안드로이드용 모바일 DJI SDK의 비행 경로 제어는 다음과 같이 세가지 방법을 제공한다.
- 수동 : 사용자가 리모콘으로 드론을 조종하는 동안 SDK는 실시간 비디오 및 센서 데이터를 모니터링
- 가상 스틱 명령 : SDK를 사용하면 드론 비행을 시뮬레이트하여 원격 컨트롤러 스틱 동작을 가상으로 생성
- 임무 : 드론의 제어를 용이하게 하며 미리 정의된 비행경로는 웨이 포인트 미션으로 실행
- DJI SDK와 모바일 앱의 연동은 다음과 같은 다이어그램으로 나타낼 수 있다.



[DJI SDK와 모바일앱의 연동]

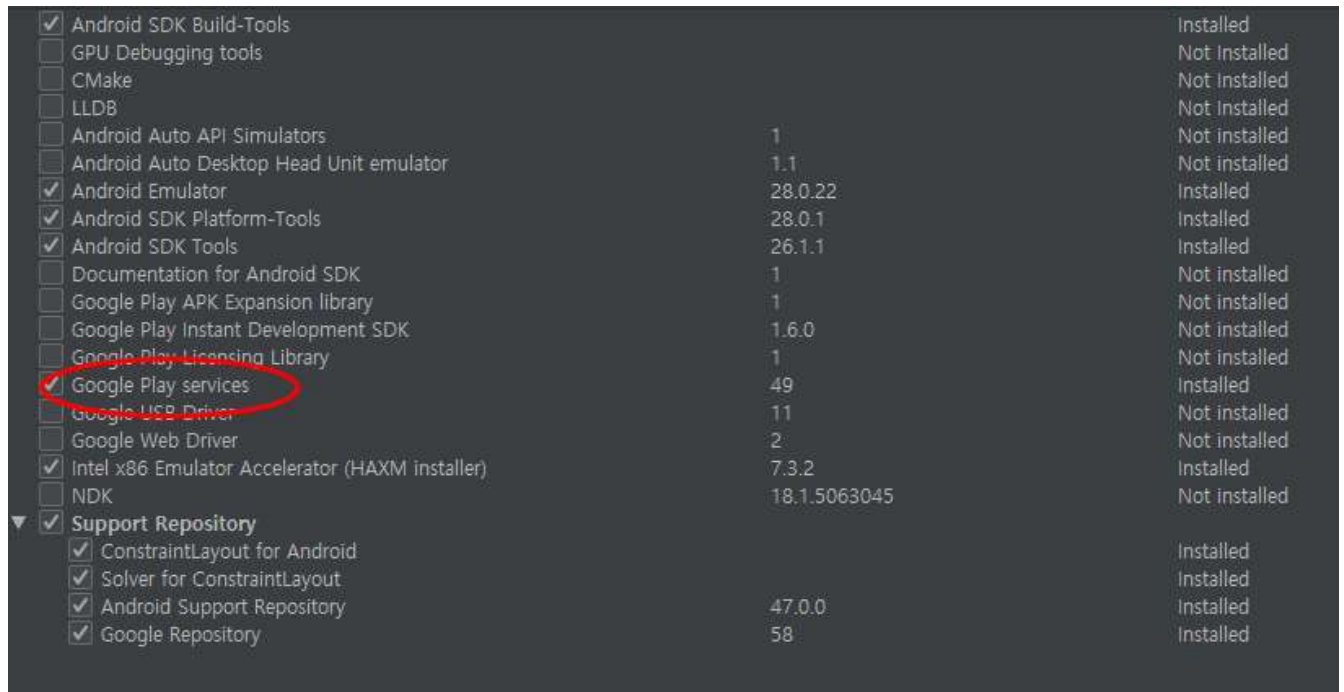
- 안드로이드용 모바일 DJI SDK 다음과 같이 구성되어 있다.

- DJI 제품에 액세스 할 수있는 라이브러리 / 프레임워크
- 항공기 시뮬레이터 및 시각화 도구
- iOS 용 디버거 및 원격 로거
- 샘플 코드 및 자습서
- 이 개발자 가이드 및 API 설명서

○ Google Maps 안드로이드 API 적용

- 앱을 제작하기 위한 순서는 다음과 같다.
 - 새 프로젝트를 선택하여 "Waypoint"라는 새 프로젝트를 만든다.
 - API 19: Android 4.4 (KitKat) 의 "Phone and Tablet"에 대해 최소 SDK 버전을 설정한다.
 - "MainActivity"로, 레이아웃 이름을 "activity_main"으로하고 프로젝트를 작성한다.
 - 구글 맵을 사용하기 위해 구글 플레이 서비스에서 해당 사이트의 절차에 따라 구글맵 API 인증키를 부여받아야 한다. 이 인증키는 앱 개발시 이용되므로 반드시 필요하다.

- 인증키까지 부여 받고 앱의 프로젝트에서 Google Play Service SDK를 설치 한다.



[SDK 관리자에서 Google Play Service 설치]

- Google지도 Android API 는 Google Play 서비스 패키지에 포함되어 있으므로Android SDK 관리자 를 사용하여 Google Play 서비스 SDK를 설치하여야 한다.
- Google Play 서비스 패키지가 설치된 후 앱의 AndroidManifest.xml 파일을 열고 다음 요소를 <application> 요소의 하위 요소로 추가한다.

```
<meta-data android:name="com.google.android.gms.version"
    android:value="@integer/google_play_services_version" />
```

```
<meta-data
    android:name="com.google.android.geo.API_KEY"
    android:value="API 키" />
```

- API 키는 구글 플레이 서비스에서 부여받은 키값을 입력하면 된다.
- 다음으로 build.gradle에 다음을 추가한다.

```
implementation 'com.google.android.gms:play-services-maps:16.0.0'
implementation 'com.google.android.gms:play-services-location:16.0.0'
```

- 그다음 AndroidManifest.xml에 퍼미션 권한을 추가 한다.

```
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
```

```

<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW" />
<uses-permission android:name="android.permission.VIBRATE" />
<uses-permission android:name="android.permission.WAKE_LOCK" />

```

- 프로젝트에서 구글 플레이 서비스를 사용하려면 Multidex 지원을 추가하여 Gradle의 64K 제한을 없애기 위해 다음과 같이 지원 라이브러리를 포함 한다.

```

compileSdkVersion 28
defaultConfig {
    applicationId "com.flagon8b.waypoint"
    minSdkVersion 15
    targetSdkVersion 28
    versionCode 1
    versionName "1.0"
    testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
}

```

- 위의 설정에서 "compileSdkVersion", "buildToolsVersion", "minSdkVersion" 및
- "targetSdkVersion"을 설정한다음 도구 -> Android -> Gradle Files이 있는 프로젝트 동기화를 선택하여 gradle 파일을 동기화한다.

• Google Maps 구현

- 구글 맵은 프래그먼트에서 호출되며 이는 리소스의 레이아웃에서 지정을 한다.

```

<fragment
    android:id="@+id/map"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    class="com.google.android.gms.maps.SupportMapFragment" />

```

- 메인 코드는 비교적 간단하여 onCreate에 다음을 추가 하면 화면에 구글맵이 호출된다.

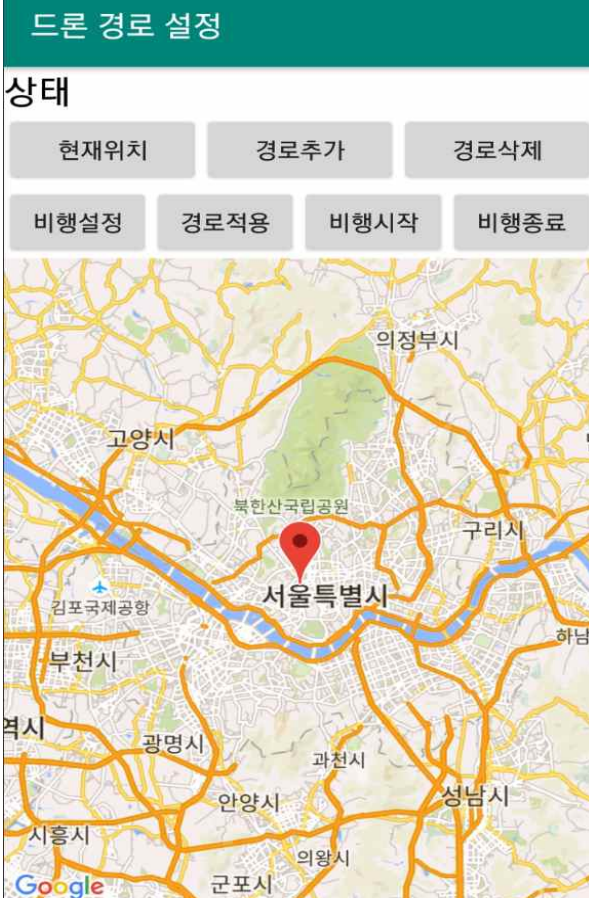
```

val mapFragment = getSupportFragmentManager()
    .findFragmentById(R.id.map) as SupportMapFragment
mapFragment.getMapAsync { map ->
    val SEOUL = LatLng(37.56, 126.97)
    val markerOptions = MarkerOptions()
    markerOptions.position(SEOUL)
    markerOptions.title("서울")
    markerOptions.snippet("한국의수도")
    map!!.addMarker(markerOptions)
    map!!.moveCamera(CameraUpdateFactory.newLatLng(SEOUL))
    map!!.animateCamera(CameraUpdateFactory.zoomTo(10f))
}

```

- DJI SDK의 비행 관련 클래스

Manager Classes	DJISDKManager
	KeyManager
	FlyZoneManager
	AppActivationManager
	UserAccountManager
	DataProtectionManager
	LDManager
	FlightHubManager
Base Classes	BaseProduct
	BaseComponent
Product Classes	Aircraft
	HandHeld
Component Classes	FlightController
	Battery
	Camera
	Gimbal
	AirLink
	RemoteController
	HandheldController
	MobileRemoteController
	Payload
	AccessoryAggregation
Mission Classes	MissionControl
	MissionAction
	TimelineMission
	Trigger
	WaypointMissionOperator
	FollowMeMissionOperator
	PanoramaMissionOperator
	HotpointMissionOperator
	TapFlyMissionOperator
	ActiveTrackOperator

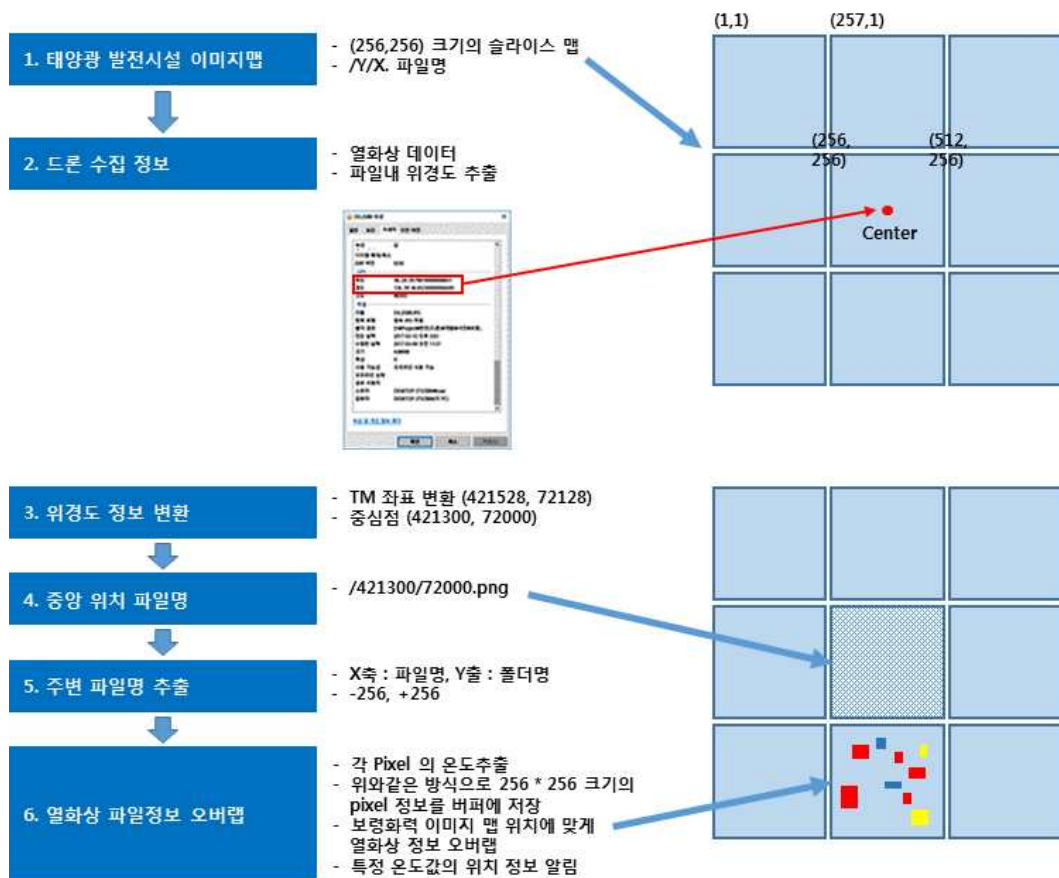
	상태 : 드론의 현재 상태를 표출 현재위치 : 드론의 현재 위치를 표시 경로추가 : 드론이 비행할 경로를 포인트로 추가 경로삭제 : 현재 지정한 경로를 삭제 비행설정 : 비행 시 드론의 상태를 설정 경로적용 : 드론으로 비행경로 설정을 입력 비행시작 : 드론 비행 시작 비행종료 : 비행을 끝냄
--	--

	고도 : 드론의 비행 고도 설정 속도 : 드론의 비행 속도 지정 비행후 작동 : 비행 후 드론의 작동 방법을 지정 헤딩 : 비행 시 드론의 전방 방향을 지정
---	---

(3) 무선 열화상 카메라와 연동된 시설물 상태 분석 시스템 개발

○ 열화상 정보를 이용한 시설물 상태 분석 알고리즘 구현

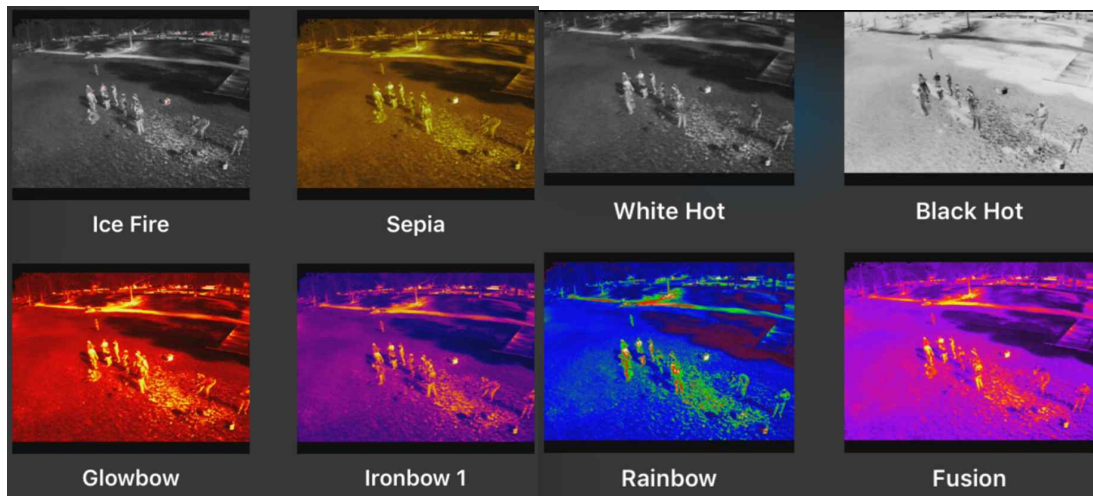
- 태양광 발전 시설 상태 모니터링 SW를 개발하기 위해서 열화상 정보를 이용한 시설물 상태 체크 모듈을 개발하였다.
- 이를 위해 열화상 카메라 촬영 이미지 등을 활용하여 시설물 상태 검토 SW개발을 위해서는 1) 대상 시설물의 이미지 맵 작성 및 시설물의 열화상 데이터 수집하고 2) 위치기반 Image Map과 드론을 통한 수집 정보의 위치 매핑 모듈을 접목하였다.
- 다음은 시설 상태 모니터링 SW 개발 프로세스로서 수집된 열화상 정보를 분석, 대상 시설물의 상태를 확인/검토할 수 있도록 구성된 프로세스이다.



○ 영상도화 이미지 맵과 수집된 열화상 정보와의 매핑 기술 및 검출 기능 개발

• 열화상 카메라 특성

- 열화상 카메라로 촬영된 데이터는 전체 이미지가 각각의 Pixel의 온도에 따라 고유의 색상을 가지고 있어 전체 이미지에 모두 색을 가지고 있으며 다음과 같이 여러 가지 모드가 있다.



[열화상 카메라 모드]

- 여러 모드 중 White Hot의 경우 특정 온도 검출에 적합하고 Rainbow의 경우 온도분포 검출에 적합하다. 즉,

- Rainbow 모드 : 온도분포 측정
- White Hot 모드 : 특정온도 정밀검색 측정

- 온도분포 모드 특징

- 촬영 지역의 전반적인 온도분포를 색깔을 통해 구분가능
- 각 촬영 주변의 온도변화를 빠르게 파악
- 최고온도, 최저온도 그리고 평균온도 확인가능
- 시간별 복사열에 의한 온도변화의 온도분포 확인가능
- 계절별 온도변화에 따른 온도분포 확인가능

- 특정온도 정밀검색

- 넓은 지역에 분포한 특정온도 지점들 예측
- 특정온도로 예상되는 위치파악 가능
- 자료의 Data Base화하여 수치분석 가능

○ 특정온도 검출을 위한 이미지 분석

• 이미지 촬영 및 분석 단계

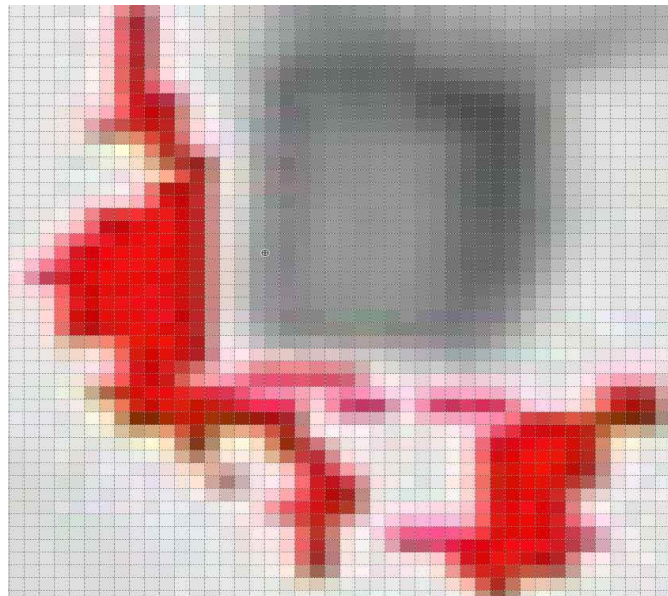
- 열화상 카메라로 촬영한 원본 데이터를 시스템에 복사
- 온도분포 데이터와 특정온도 데이터를 구분하여 복사
- 온도분포 이미지를 화면에 표시
- 특정온도 이미지를 화면에 표시하고 특정온도를 검출하여 해당 지점 표시



[이미지 분석 단계]

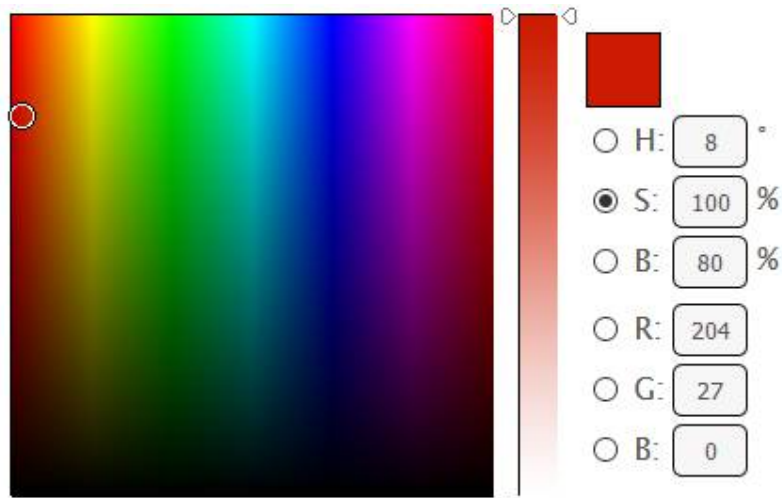
- 열화상 카메라 이미지 분석

- 카메라의 SD카드에 저장된 이미지를 선택하여 시스템의 특정 폴더로 복사한다.
- 촬영이미지에 저장되어 있는 위경도 좌표의 정확한 픽셀위치(X값, Y값)가 정해져 있지 않기 때문에 약간의 오차가 발생한다.
- 특정 온도 이미지를 선택하여 각 이미지에 대한 온도 분석을 시작하며 특정 온도 색상이 발견된 이미지는 온도 분포 이미지와 같은 방식으로 해당 위치에 표시를 한다.
- 온도 이미지가 특정 온도 이상의 색상은 붉은색 계통으로 표시하지만 시스템에서 표현할 수 있는 붉은색 계통은 몇십만 가지가 넘으며 이를 감지하기 위해서는 사람의 눈에 붉게 보이는 색상 분포도의 패턴을 찾아내는 알고리즘을 개발하였다.



[여러 가지 붉은색 패턴]

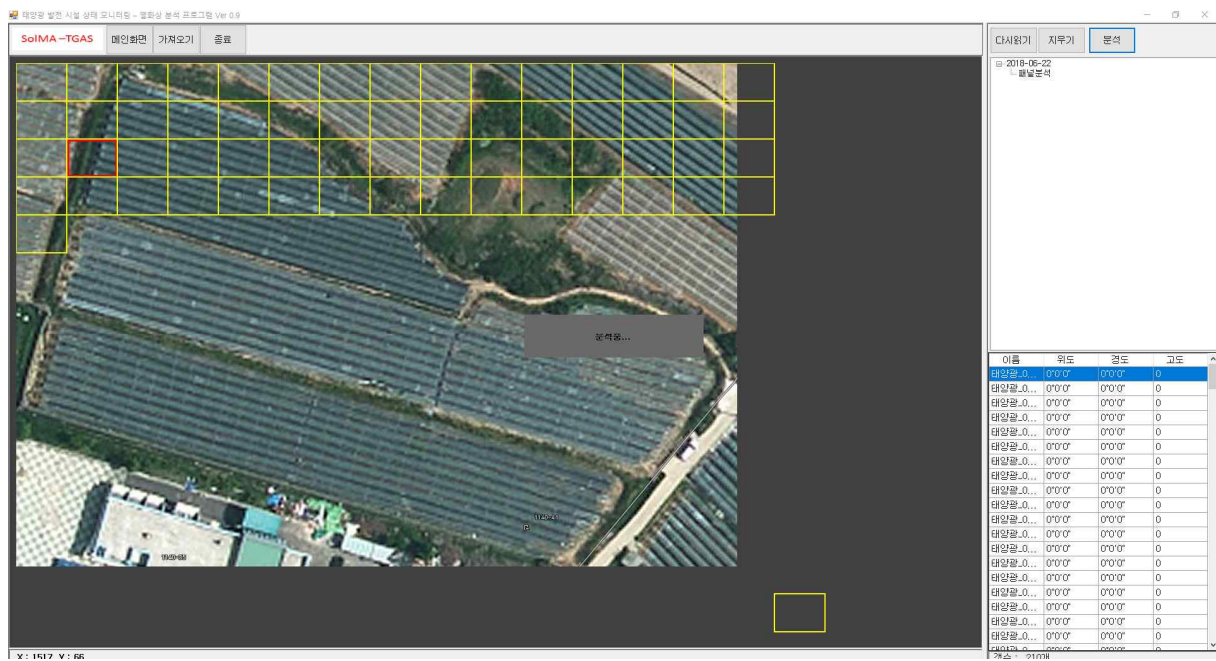
- 컴퓨터에서 색상은 RGB로 표현되며 그 의미는 빛의 삼원색인 Red, Green, Blue의 조합으로 각 요소는 8bit이다.
- 즉, 각각의 색은 각각 2의 8승인 256 가지의 값을 가질 수 있으며 빛의 삼원색인 RGB가 표현할 수 있는 색의 총 가짓수는 $256 \times 256 \times 256$ 즉, 16,777,216 가지의 색을 표현할 수 있다.



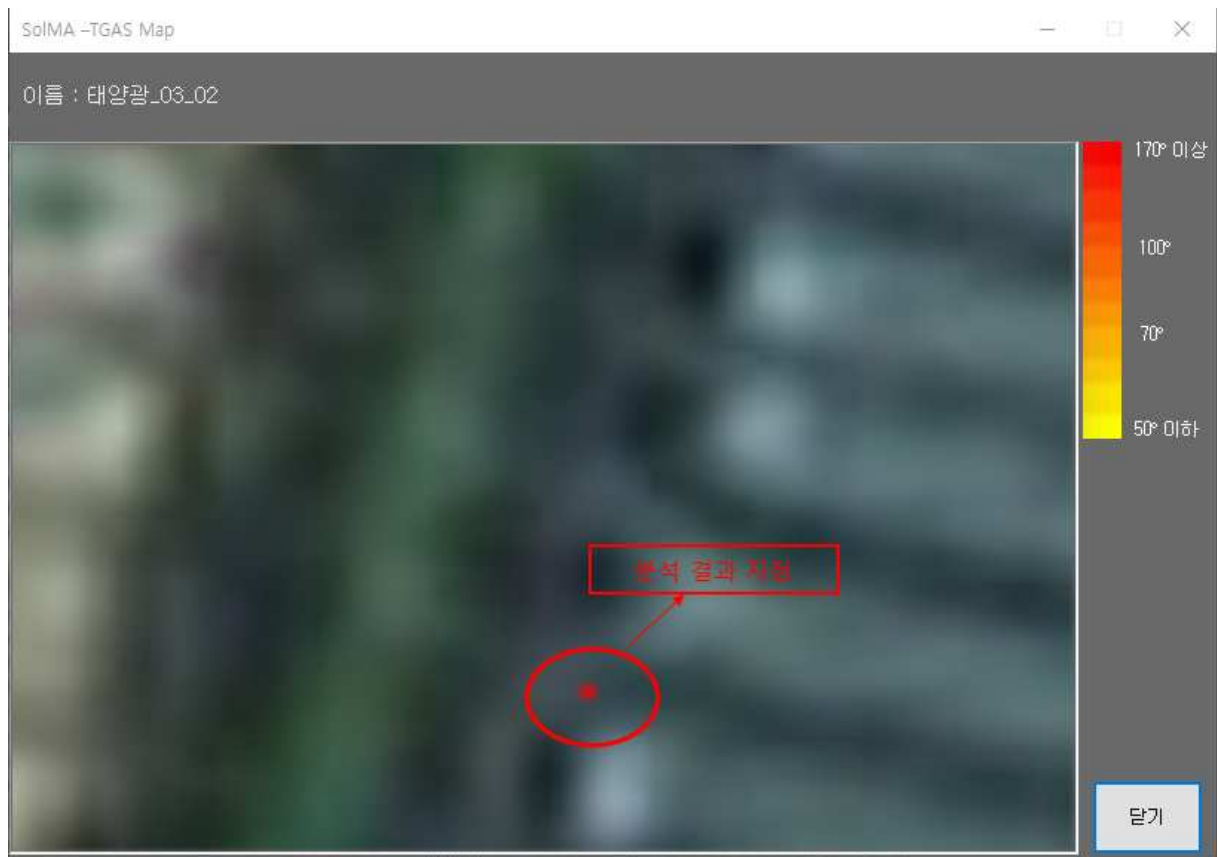
[RGB 색상 패턴의 Red 팔레트]

- 태양광 발전 시설 상태 검토 기능

- 태양광 발전 시설 지역의 이미지 데이터를 화면에 출력 후 특정 상태 검토를 하면 비행 경로 즉, 촬영 경로별 이미지를 차례로 검색하고 그 위치를 노란색 사각형으로 실시간 표시한다. 찾아낸 특정 온도 지점은 붉은색 사각형으로 표시하며 촬영이미지의 시간차에 따라 중복하여 찾아 낼 수도 있다.



[열화상 분석]



[분석 결과 상세 보기]

- 특정 상태 지점은 각 화소의 온도 분포 색상값을 읽어 들어 분석하며 빠르게 분석하기 위해 이미지의 모든 화소를 메모리에 로딩한다. 분석 후 리소스를 절약하기 위해 해당 이미지는 메모리에서 삭제된다.

5. 재난·재해(지진 및 수재해 등)용 통합 정보 플랫폼과의 효과적인 연계를 위한 시설물 감시 정보 인터페이스 모듈 구현

최근 지진, 홍수, 태풍, 가뭄 등의 자연재난과 화재, 수재해 등의 각종 사회 재난·재해가 지속적으로 증가하고 있어 재난·재해의 효과적인 예방과 대응을 위한 예측, 확산, 피해 범위 분석 등을 지원하기 위한 다양한 연구가 진행되고 있음. 이를 위하여 재난·재해용 통합 정보 플랫폼에 대한 건물 시설물에 대한 공간 및 시설 정보를 식별하고, 식별된 공간/시설 정보 가공 및 운용 모듈, 분석시스템과의 인터페이스 모듈을 개발하여 재난·재해 통합정보 플랫폼에서 효과적인 재난·재해 대응을 지원하기 위한 모듈을 구현.

- 재난·재해 분석 지원을 위한 공간정보 식별 및 인터페이스 설계
 - 화재, 지진, 침수 관련 재난·재해 분석 시스템과 정보 연계를 위한 공간 및 시설 데이터 표준 정의 및 인터페이스 설계
- 재난·재해 분석시스템 지원을 위한 공간정보 가공 및 구조화된 운용 모듈 개발
 - 플랫폼 확장을 고려한 표준화된 공간정보 연계를 지원하고, 효과적인 분석지원을 위해 구조화된 시설 데이터 정의 및 변환 기능 개발
 - 지진 및 지반 변형에 따른 건축물 피해 분석에 필요한 지반, 건축물 구조 등의 공간정보 가공 및 저장 기능 개발
 - 화재 관련 피해 예측 시스템 지원을 위한 건축물 및 재질 정보 운용 구조 개발 및 가공 기술 개발
 - 복합 시설 지하공간의 다층 연계 침수 분석을 위한 지하공간정보 가공 및 저장 기능 개발
- 재난재해(화재, 지진, 침수) 대응을 위한 분석시스템과 통합 정보 플랫폼과 효과적인 연계를 위한 인터페이스 모듈 개발
 - 개방형 통합 플랫폼과 화재, 지진, 침수 분석시스템에 필요한 공간/시설 정보를 연계할 수 있는 인터페이스 모듈 개발

(1) 재난·재해 분석 지원을 위한 재난(화재, 지진, 침수)별 요구사항 정의

① 화재 부분 요구사항

- 기둥, 보, 벽체 등의 구조 정보가 필요하며, 특히 공동구와 유사한 수직관통로가 있을 경우 반드시 표기.
- 문, 창문, 계단, 엘리베이터와 같은 건축 정보는 정확한 위치 정보가 표현.
- 화재 센서의 종류와 위치에 대한 정보가 필요.
- 위 데이터를 CAD 형태(DXF, DWG)로 제공하면 이를 기초로 모델링을 진행.
- 시간, 연기농도, 온도 등의 센싱 데이터를 이용하여 위치(방화구획No, 위험도(연기높이))를 예측할 수 있어야 함.

② 지진 부분 요구사항

- 센서 : 지진가속도계(건물, 지반용), 침수계, 지하수위계, 침하계, GPS, 풍속풍향계, 경사계, 변형률계 등의 센서 필요.
- 지반정보 : 표준관입시험도(SPT), 주상도가 필요. 이때, 세부 물성정보는 토질과 암반으로 구분하여 제공.
- 기본 구조 데이터 외에 추가적으로 재질에 대한 정보가 필요. 여기서의 재질은 두께와 구조에 사용된 재질의 강도 등의 추가 정보를 포함.

③ 침수 부분 요구사항

- 기초, 기둥, 보, 벽체, 계단, 경사로, 공간(층, 복도, 룸)의 정보를 TecPlot (Node, Element 구조)의 ASCII 형태로의 제공이 필요.
- 하수관, 배수펌프, 지표고도의 추가 정보도 분석에 필요.

(2) 재난·재해 분석 알고리즘에서 필요한 데이터 양식 조사

① IFC

RVT, DWG 등 독점적인 파일 형식의 경우에는 특정 소프트웨어에서만 사용할 수 있도록 각 벤더사에 따라 형식이 다르고 사용자가 확인이 불가하지만 중립적인 비독점적인 파일 형식인 IFC 파일은 텍스트 형태의 파일로 사용자가 해당 정보를 확인 가능. IFC파일의 실제 모습은 다음과 같다.

```

ISO-10303-21;
HEADER;
FILE_DESCRIPTION (('ViewDefinition [CoordinationView,
QuantityTakeOffAddOnView]'), '2;1');
FILE_NAME ('example.ifc', '2011-03-08T17:36:19', ('Architect'), ('Building
Designer Office'), 'IFC Engine DLL version 1.02 beta', 'IFC Engine DLL
version 1.02 beta', 'The authorising person');
FILE_SCHEMA (('IFC2X3'));
ENDSEC;
DATA;
#1 = IFCPROJECT('14GX40SX9EnQvslTXeHCzA', #2, 'Default Project',
'Description of Default Project', $, $, $, (#20), #7);
#2 = IFCOWNERHISTORY(#3, #6, $, .ADDED., $, $, $, 1299573379);
#3 = IFCPERSONANDORGANIZATION(#4, #5, $);
#4 = IFCPERSON('ID001', 'Bonsma', 'Peter', $, $, $, $, $);
#5 = IFCORGANIZATION($, 'TNO', 'TNO Building Innovation', $, $);
... 중략 ...
#45 = IFCWALL('3ZuNN_FnPDYQ9hY0$m2512', #2, 'Wall xyz', 'Description of
Wall', $, #46, #51, $);
#46 = IFCLOCALPLACEMENT(#36, #47);
#47 = IFCAXIS2PLACEMENT3D(#48, #49, #50);
#48 = IFCCARTESIANPOINT((0., 0., 0.));
#49 = IFCDIRECTION((0., 0., 1.));
#50 = IFCDIRECTION((1., 0., 0.));
#51 = IFCPRODUCTDEFINITIONSHAPE($, $, (#72, #101));
... 중략 ...
ENDSEC;
END-ISO-10303-21;

```

IFC는 위와 같이 표준코드인 “ISO-10303-21”로 시작되며 크게 헤더(HEADER) 영역과 데이터(DATA) 영역으로 구분된다. 헤더 영역은 “HEADER”라는 문자로 시작되며 파일의 설명, 명칭, 스키마 등 해당 파일의 정보를 설명하고, 데이터 영역은 “DATA”라는 문자로 시작되며 실제 건물의 모델링 정보를 표현한다. 각 영역별로 종료되는 시점에 “ENDSEC”이라는 문자로 영역별 정보 정의가 종료됨을 정의한다. 마지막으로 “END-ISO-10303-21”로 파일의 종료를 정의한다. 데이터 영역에 정의되는 각 객체의 정의 및 속성정보 할당 규칙은 buildingSMART¹⁾의 레퍼런스에 명시되어 있다.

② DXF

DXF 파일은 서로 다른 컴퓨터 지원 설계(CAD) 프로그램 간에 설계 도면 파일을 교환하는 데 업계 표준으로 사용되는 파일 형식. 도면 교환 형식이라는 뜻이다. 처음에는 미국 오토 데스크(Auto Desk)사가 자사의 CAD 프로그램인 오토 캐드(Auto CAD)의 외부 파일 형식으로 개발하였으나, 오토 캐드가 세계적으로 많이 판매되었기 때문에 다른 CAD 프로그램도 DXF를 채택하지

1) 건설 프로젝트의 전 생명주기의 정보 공유를 위한 산업표준 즉 IFC(Industry Foundation Classes)를 제정한 건설 정보 공유 및 교환을 위하여 AEC/FM 분야의 산업기술기반의 표준 통합을 목적으로 1994년 9월 결성된 국제적인 비영리단체.

않을 수 없게 되었으며, 오토 데스크사에서도 적극적으로 그 규격을 공개했다. 그 결과 DXF가 다양한 CAD 프로그램 간의 도면 교환에 널리 사용되었다. 현재 미국에서는 업계 표준과 같이 되었으며, 한국 등 많은 나라에서 표준에 가까운 위치를 차지하게 되었다.

```

0
SECTION
2
HEADER
9
$ACADVER
1
AC1015
9
$ACADMAINTVER
70
20
9
$DWGCODEPAGE
3
ANSI_949
9
... 중략 ...
ObjectDBX Classes
90
0
280
0
281
0
0
CLASS
1
... 중략 ...
AcDbXrecord
280
1
102
ENDSEC
0
EOF

```

- 파일의 구조

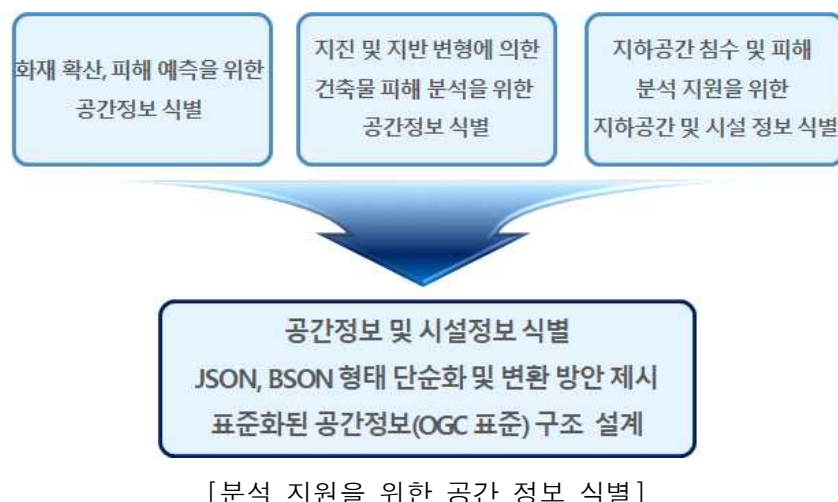
- HEADER SECTION : 도면에 대한 일반 정보를 포함한다. 이 영역은 AutoCAD 데이터베이스의 버전 번호와 다양한 시스템 변수로 구성된다. 각 매개변수는 변수 이름과 연관된 값을 포함하고 있다.
- CLASSES SECTION : 데이터베이스의 BLOCKS, ENTITIES 및 OBJECTS 섹션에서 인스턴스가

사용되는 응용프로그램 정의 클래스에 대한 정보를 보유하고 있다. 클래스 정의는 클래스 계층에 고정된다.

- TABLES SECTION : 다음의 기호 테이블에 대한 정의를 포함한다.
 - APPID(응용프로그램 식별 테이블)
 - BLOCK_RECORD(블록 참조 테이블)
 - DIMSTYLE(치수 스타일 테이블)
 - LAYER(도면층 테이블)
 - LTYPE(선종류 테이블)
 - STYLE(텍스트 스타일 테이블)
 - UCS(사용자 좌표계 테이블)
 - VIEW(View table,테이블 보기)
 - VPORT(뷰포트 구성 테이블)
- BLOCKS SECTION : 도면의 각 블록 참조를 구성하는 블록 정의 및 도면요소(entity- 점, 선, 면, 원, 텍스트 등 도면을 형성하는 각 그림 요소를 의미한다)를 포함한다.
- ENTITIES SECTION : 블록 참조(삽입 도면요소)와 함께 도면의 그래픽 객체(도면요소)를 포함한다.
- OBJECTS SECTION : 도면의 비그래픽 객체를 포함한다. 도면요소 또는 기호 테이블 레코드 또는 기호 테이블이 아닌 객체는 모두 이 섹션에 저장된다. OBJECTS 섹션에 표시되는 항목의 예로는 다중선 스타일 및 그룹이 포함된 라이브러리를 들 수 있다.
- THUMBNAILIMAGE SECTION : 도면의 미리보기 이미지 데이터를 포함한다. 이 섹션은 선택 사항이다.
- END OF FILE : 파일의 끝

(3) 공간 및 시설 정보 등을 각 분석시스템에 전달하기 위한 구조화된 데이터 정의 및 확보

재난·재해 분석을 위해 필요한 공간 및 시설정보를 효과적으로 연계하기 위해 복합시설물의 공간 정보를 단순화 및 표준화 작업 수행.



(4) 분석 연계 인터페이스 설계

지진은 크게 층간변위, 층전단력, 보, 기둥 및 벽체에 대한 분석 데이터가 필요하며, 침수는 시간별 위치에 대한 침수 깊이 데이터가 필요하다. 화재의 경우에는 화재 규모별로 시간대별 연기 높이 데이터를 제공하여, 연기 전파 시간차를 이용하여 화재의 규모를 예측할 수 있도록 지원해야 한다. 이를 위해 본 연구에서는 한국건설기술연구원의 협조를 통하여 예시 값으로 다음과 같은 파일럿 자료를 이용하였다.

구분		해석 대상 데이터 파일
지진	층간변위	Story drift_raw data.txt
	층전단력	Shear_TH_X.txt
		Shear_TH_Y.txt
	보	RG5_HL__i.txt
		RG5_TH_M_i.txt
		RG5_TH_V_i.txt
		RG5_TH_R_i.txt
	기둥	5C3_TH_AS.txt
		5C3_TH_M2_j.txt
		5C3_TH_M3_j.txt
		5C3_TH_P_j.txt
		5C3_TH_R2.txt
		5C3_TH_R3.txt
		5C3_TH_V2_j.txt
		5C3_TH_V3_j.txt
	벽체	W1_1F_R.txt
		W1_1F_u_AS.txt
		W1_M.txt
		W1_P.txt
		W1_V.txt
침수	SNAPSHOT_001.PLT ~ SNAPSHOT_020.PLT	
화재	0.56MW	Building_devc.csv
		Building_hrr.csv
	2.80MW	Building_devc.csv
		Building_hrr.csv

• 층간 변위 데이터

층간변위 데이터는 txt 파일로 존재하며, 제목, 대상건물 정보, 입력하중 정보, 하중조합 등 기본적인 메타데이터와 실제 해석 값을 구분 지을 수 있는 속성(컬럼) 정보, 실제 해석 값, 해석 값에 대한 최대, 최소값 정보로 구성된다. 실제 해석 데이터를 분석한 결과는 다음과 같다.

구분	실제 데이터
메타 데이터	1 PERFORM DRIFT TIME HISTORIES

	Structure = KIST_Main (KIST main building) Analysis Series = Earthquake () Load Case =, [2] = [1] + EQ
컬럼 데이터	Column 1 =, time (sec) Column, 2, = drift no., 2, name =, mass center_5F Column, 3, = drift no., 3, name =, mass center_4F Column, 4, = drift no., 4, name =, mass center_3F Column, 5, = drift no., 5, name =, mass center_2F Column, 6, = drift no., 6, name =, mass center_5F_Y Column, 7, = drift no., 7, name =, mass center_4F_Y Column, 8, = drift no., 8, name =, mass center_3F_Y Column, 9, = drift no., 9, name =, mass center_2F_Y Column, 10, = drift no., 10, name =, LU_5F_X ...중략... Column, 20, = drift no., 20, name =, RB_5F_X Column, 21, = drift no., 21, name =, RB_4F_X Column, 22, = drift no., 22, name =, RB_3F_X Column, 23, = drift no., 23, name =, RB_2F_X Column, 24, = drift no., 24, name =, RB_1F_X Column, 25, = drift no., 25, name =, RB_5F_Y Column, 26, = drift no., 26, name =, RB_4F_Y Column, 27, = drift no., 27, name =, RB_3F_Y Column, 28, = drift no., 28, name =, RB_2F_Y Column, 29, = drift no., 29, name =, RB_1F_Y Column, 30, = drift no., 31, name =, mass center_1F Column, 31, = drift no., 32, name =, mass center_1F_Y Column, 32, = drift no., 33, name =, mass center_1-5F Column, 33, = drift no., 34, name =, mass center_1-5F_Y
해석값	0,-7.5174e-6,-2.4261e-6,-2.4308e-6,-1.3828e-6, 1.5654e-5, 1.6954e-5, 1.8895e-5, 1.7136e-5,-2.5243e-6,-1.1245e-6,-1.3484e-6,-5.3864e-7, 3.2381e-6,-5.3426e-5,-4.5701e-5,-3.2976e-5,-2.4278e-5,-1.6933e-5, 1.3018e-5, 1.2916e-5, 1.0275e-5, 8.7377e-6, 1.0825e-5, 8.5899e-5, 8.016e-5, 7.1222e-5, 5.8878e-5, 5.1074e-5, 2.6043e-6, 1.6862e-5,-2.1255e-6, 1.7095e-5 .005,-7.5174e-6,-2.4261e-6,-2.4307e-6,-1.384e-6, 1.5654e-5, 1.6954e-5, 1.8895e-5, 1.7137e-5,-2.5243e-6,-1.1245e-6,-1.3482e-6,-5.3984e-7, 3.2157e-6,-5.3426e-5,-4.5701e-5,-3.2975e-5,-2.4278e-5,-1.6924e-5, 1.3018e-5, 1.2916e-5, 1.0275e-5, 8.7366e-6, 1.0802e-5, 8.5899e-5, 8.016e-5, 7.1222e-5, 5.8879e-5, 5.1082e-5, 2.5819e-6, 1.687e-5,-2.1305e-6, 1.7097e-5 ...중략... 19.995,-6.3529e-4,-6.6704e-4,-6.9178e-4,-6.9033e-4,-8.953e-6, 1.2186e-5, 1.5716e-5, 1.4848e-5,-6.3745e-4,-6.6692e-4,-6.9275e-4,-6.9266e-4,-4.1151e-4,-5.838 8e-5, 6.5412e-6, 6.1296e-5, 1.291e-4, 1.3798e-4,-6.2642e-4,-6.6567e-4,-7.0297e-4,-7.1824e-4,-4.3704e-4, 4.0444e-5, 1.7888e-5,-3.0292e-5,-1.0026e-4,-9.0874e-5,-4.0972e-4, 2.4446e-5,-6.1428e-4, 1.1926e-5

	20,-6.4506e-4,-6.7631e-4,-6.9991e-4,-6.9696e-4,-9.1074e-6, 1.2245e-5, 1.6434e-5, 1.6336e-5,-6.4709e-4,-6.7617e-4,-7.0086e-4,-6.9925e-4,-4.1454e-4,-6.014 e-5, 5.5425e-6, 6.1169e-5, 1.2908e-4, 1.381e-4,-6.357e-4,-6.7468e-4,-7.1089e-4,-7.2449e-4,-4.3976e-4, 4.1915e-5, 1.9016e-5,-2.8723e-5,-9.726e-5,-8.8047e-5,-4.1277e-4, 2.5908e-5,-6.2156e-4, 1.2657e-5
해석 최대값	Maximum, .0053523, .005497, .0053444, .005018, .0065016, .0065788, .0061772, .0054383, .0053027, .0054844, .0053344, .0050211, .0033488, .0060778, .0063427, .0061384, .0058619, .004065, .0052052, .005421, .0052337, .0050633, .0034953, .006956, .0068355, .0063041, .0054341, .0033144, .00334, .0034321, .0047097, .0055662
해석 최소값	Minimum,-.0060056,-.0062919,-.0063896,-.0064862,-.008721,-.0087132,-.00 81566,-.0071317,-.0059279,-.0062675,-.006376,-.0064746,-.004234,-.00887 22,-.0088083,-.0080074,-.0066815,-.004363,-.0056076,-.00601,-.0062333,- .0063492,-.004261,-.0085712,-.0086172,-.0083096,-.0075962,-.0043057,-.0 042338,-.0043349,-.0057282,-.0072119

- 충전단력 데이터

충전단력 데이터는 txt 파일로 존재하며 X, Y 데이터 파일로 나뉜다. 각 파일은 제목, 대상 건물 정보, 입력하중 정보, 하중조합, 단위 등 기본적인 메타데이터와 실제 해석 값을 구분 지을 수 있는 속성(컬럼) 정보, 실제 해석 값, 해석 값에 대한 최대, 최소값 정보로 구성된다. 실제 해석 데이터를 분석한 결과는 다음과 같다.

구분	실제 데이터
메타 데이터	Story Shear_Time Histories_X Direction (H1) PERFORM STRUCTURE SECTION RESULT TIME HISTORIES Structure = KIST_0905 (KIST main building_gage, Structure Section 추) Analysis Series = Earthquake () Load Case =, [2] = [1] + EQ Units =, kN, m
컬럼 데이터	Column 1 =, time (sec) Column, 2, = section no., 13, name =, 5F Column, 3, = section no., 14, name =, 4F Column, 4, = section no., 15, name =, 3F Column, 5, = section no., 16, name =, 2F Column, 6, = section no., 17, name =, 1F Column, 7, = section no., 18, name =, B1F
해석값	0, 1.4424e-5, 1.0014e-5, 1.058e-5, 4.7684e-6, 4.1723e-6, 4.53e-6 .005, .0084072, .019829, .032439, .16172,-.46197,-.46197 .01, .043837, .092209, .19192, .60289,-2.0136,-2.0135

	.015, .11869, .24101, .58215, 1.0727, -4.6432, -4.6433 .02, .24047, .50714, 1.2512, 1.1627, -7.9878, -7.9877 ...중략... 19.98, -475.88, -775.06, -1007.1, -1191.7, -1352.5, -1352.5 19.985, -509.32, -805.94, -1016.7, -1185.4, -1342.8, -1342.8 19.99, -540.46, -832.94, -1026, -1180.3, -1335, -1335 19.995, -568.75, -856.55, -1036.6, -1180.8, -1329.7, -1329.7 20, -591.66, -876, -1048.3, -1183, -1327.4, -1327.4
해석 최대값	Maximum, 6025.5, 7873.4, 8973.5, 10620, 12007, 12007
해석 최소값	Minimum, -5815.1, -8447.4, -10262, -12756, -13676, -13676

구분	실제 데이터
메타 데이터	Story Shear_Time Histories_Y Direction (H2) PERFORM STRUCTURE SECTION RESULT TIME HISTORIES Structure = KIST_0905 (KIST main building_gage, Structure Section 추) Analysis Series = Earthquake () Load Case =, [2] = [1] + EQ Units =, kN, m
컬럼 데이터	Column 1 =, time (sec) Column, 2, = section no., 13, name =, 5F Column, 3, = section no., 14, name =, 4F Column, 4, = section no., 15, name =, 3F Column, 5, = section no., 16, name =, 2F Column, 6, = section no., 17, name =, 1F Column, 7, = section no., 18, name =, B1F
해석값	0, 1.0427e-5, -3.1833e-5, 7.4552e-5, -8.7392e-5, 6.6306e-5, -3.5409e-6 .005, -.0086432, -.014217, -.023675, -.06212, .22995, .22997 .01, -.042321, -.06796, -.11982, -.21578, 1.0025, 1.0024 .015, -.11179, -.18066, -.32267, -.32748, 2.3149, 2.3149 .02, -.22634, -.37514, -.62885, -.19992, 3.9924, 3.9926 ...중략... 19.98, -1.2238, 15.45, 46.651, 93.173, 160.97, 160.97 19.985, -14.676, 2.6487, 44.476, 108.08, 188.14, 188.14 19.99, -27.076, -9.463, 43.193, 123.33, 214.18, 214.18 19.995, -38.58, -20.807, 42.364, 137.22, 237.67, 237.67 20, -48.338, -29.609, 43.139, 150.44, 257.34, 257.34
해석 최대값	Maximum, 7646.4, 10115, 11422, 13240, 15995, 15995
해석 최소값	Minimum, -7390.2, -10095, -11448, -14021, -16815, -16815

- 보 데이터

보 데이터는 txt 파일로 존재하며 크게 Histresis Loop파일과 Time Histories파일로 나뉜다. 각 파일에는 제목, 대상건물 정보, 입력하중 정보, 하중조합 등 기본적인 메타데이터와 실제 해석 값 정보로 구성된다. 해석 값 정보는 모든 데이터가 두 개의 필드를 갖으며 첫 번째 필드의 정보는 시간, 두 번째 필드의 정보는 해석 값을 나타낸다, Histresis Loop파일과 Time Histories파일은 같은 구조로 구성되어 있다. 실제 해석 데이터를 분석한 결과는 다음 표와 같다.

구분	실제 데이터
메타 데이터	RG5_i단_Histresis Loops PERFORM HYSTERESIS LOOPS Structure = KIST_Main (KIST main building) Analysis Series = Earthquake () Load Case = [2] = [1] + EQ
해석값	0,5.048501E-04 0.005,5.048501E-04 0.01,5.048501E-04 0.015,5.048501E-04 0.02,5.048498E-04 0.025,5.048492E-04 0.03,5.048477E-04 0.035,5.048441E-04 0.04,5.048375E-04 ...중략... 19.93954,4.616421E-04 19.94454,4.651102E-04 19.94954,4.684122E-04 19.96954,4.797408E-04 19.97454,4.820538E-04 19.97954,4.841292E-04 19.98454,4.859433E-04 19.98954,4.874681E-04 19.99454,4.886761E-04 19.99953,4.895508E-04

- 기동 데이터

기동 데이터는 txt 파일로 존재하며, 각 파일에는 제목, 대상건물 정보, 입력하중 정보, 하중조합 등 기본적인 메타데이터와 실제 해석 값 정보로 구성된다. 해석 값 정보는 모든 데이터가 두 개의 필드를 갖으며 첫 번째 필드의 정보는 시간, 두 번째 필드의 정보는 해석 값을 나타낸다, 실제 해석 데이터를 분석한 결과는 다음과 같다.

구분	실제 데이터
메타 데이터	5C3_Time Histories _Axial Strain PERFORM TIME HISTORY Structure = KIST_0905 (KIST main building_gage,Structure Section 추) Analysis Series = Earthquake () Load Case = [2] = [1] + EQ
해석값	0,-3.903728E-05 0.005,-3.903728E-05 0.01,-3.903728E-05 0.015,-3.903728E-05 0.02,-3.903728E-05 0.025,-3.903729E-05 0.03,-3.90373E-05 0.035,-3.903731E-05 0.04,-3.903732E-05 ...중략... 19.94454,-3.976138E-05 19.94954,-3.974353E-05 19.95454,-3.972447E-05 19.95954,-3.97044E-05 19.96454,-3.968357E-05 19.96954,-3.966225E-05 19.97454,-3.96407E-05 19.97954,-3.961926E-05 19.98454,-3.959825E-05 19.98954,-3.957804E-05 19.99454,-3.955897E-05 19.99953,-3.954141E-05

- 벽체 데이터

벽체 데이터는 txt 파일로 존재하며 두 가지 구조로 데이터가 구성되어있다. 제목, 대상건물 정보, 입력하중 정보, 하중조합 등 기본적인 메타데이터와 실제 해석 값 정보로 구성되는 구조와 기본적인 메타데이터와 실제 해석 값을 구분 지을 수 있는 속성(컬럼) 정보, 실제 해석 값, 해석 값에 대한 최대, 최소값 정보로 구성되는 구조로 나뉜다. 실제 해석 데이터를 분석한 결과는 다음 표와 같다.

구분	실제 데이터
메타 데이터	W1_B1~5F_Moment about X_Time Histories PERFORM STRUCTURE SECTION RESULT TIME HISTORIES Structure = KIST_0905 (KIST main building_gage,Structure Section 추) Analysis Series = Earthquake () Load Case =, [2] = [1] + EQ Units =, kN, m
컬럼 데이터	Column 1 =, time (sec) Column, 2, = section no., 7, name =, W1_5F Column, 3, = section no., 8, name =, W1_4F Column, 4, = section no., 9, name =, W1_3F Column, 5, = section no., 10, name =, W1_2F Column, 6, = section no., 11, name =, W1_1F Column, 7, = section no., 12, name =, W1_B1F
해석값	0, .29857, 23.13, 5.0579, 7.3935, .58141, 4.8445 .005, .30337, 23.14, 5.0619, 7.3989, .51305, 4.8275 .01, .32223, 23.173, 5.0727, 7.4085, .27959, 4.7699 .015, .36132, 23.231, 5.0895, 7.4027,-.13391, 4.6702 .02, .42503, 23.311, 5.1106, 7.3523,-.70044, 4.5388 ...종락... 19.98, 132.77,-19.346,-152.1,-261.01,-219.49, 47.062 19.985, 135.34,-14.988,-147.47,-260.19,-226.63, 44.823 19.99, 137.45,-11.456,-144.18,-259.92,-232.17, 42.666 19.995, 139.05,-8.9853,-142.52,-259.87,-236.23, 40.752 20, 139.94,-7.7863,-142.45,-260.77,-238, 39.192
해석 최대값	Maximum, 770.94, 659.69, 1606.3, 1128.3, 2764.6, 1041
해석 최소값	Minimum,-1670.4,-3571.2,-4462.9,-6885.9,-8859.3,-1728.6

- 화재 분석 데이터

화재 데이터는 csv 형식 파일이고 *_devc, *_hrr와 같이 접미사가 붙은 파일 두 개가 한 세트
 트로 존재한다. 각 파일에는 단위, 컬럼 정보, 값으로 구성되어 있다. 실제 해석 데이터를 분
 석한 결과는 각각 다음 표와 같다.

구분	실제 데이터
단위 데이터	s,C,C,C,C,C,C,C,C,C,C,C,C,C,C,C,C,C,C,C,C,C,C,C,C,C,C,C,C,m,m, m,m,m,m,kg/kg,kg/kg,kg/kg,kg/kg,kg/kg,kg/kg,mol/mol,mol/mol,mol/mo l,mol/mol,mol/mol,mol/mol,,,,,,C,C,C,C,C,C
컬럼 데이터	Time,"TC_kit_1","TC_kit_2","TC_kit_3","TC_kit_4","TC_kit_5","TC_sm R_1","TC_smR_2","TC_smR_3","TC_smR_4","TC_smR_5","TC_laR_1","TC_la R_2","TC_laR_3","TC_laR_4","TC_laR_5","TC_stR_1","TC_stR_2","TC_st R_3","TC_stR_4","TC_stR_5","TC_ent_1","TC_ent_2","TC_ent_3","TC_en t_4","TC_ent_5","TC_liv_1","TC_liv_2","TC_liv_3","TC_liv_4","TC_li v_5","LH_kit","LH_smR","LH_laR","LH_stR","LH_ent","LH_liv","MF_kit ","MF_smR","MF_laR","MF_stR","MF_ent","MF_liv","VF_kit","VF_smR"," VF_laR","VF_stR","VF_ent","VF_liv","AVF_kit","AVF_smR","AVF_laR"," AVF_stR","AVF_ent","AVF_liv","baloconi3","large_room3","small_room 3","living_room3","kitchen3","rest_room3"
해석값	0.0000000E+000, 1.9999997E+001, 1.9999997E+001, 1.9999997E+001, 1.9999997E+001, 1.9999998E+001, 2.5000000E+000, 2.5000000E+000, 2.5000000E+000, 2.5000000E+000, 2.5000000E+000, 2.5000000E+000, 0.0000000E+000, 0.0000000E+000, 0.0000000E+000, 0.0000000E+000, 0.0000000E+000, 0.0000000E+000, 0.0000000E+000, 0.0000000E+000, 0.0000000E+000, 0.0000000E+000, 0.0000000E+000, 0.0000000E+000, 0.0000000E+000, 0.0000000E+000, 0.0000000E+000, 0.0000000E+000, 0.0000000E+000, 0.0000000E+000, 2.0000000E+001, 2.0000000E+001, 2.0000000E+001, 2.0000000E+001, 2.0000000E+001, 2.0000000E+001 ...중략... 3.5990806E+003, 2.3185604E+001, 2.4329216E+001, 2.6449486E+001, 3.2093389E+001, 3.7647150E+001, 2.1937301E+001, 2.2692965E+001, 2.4737238E+001, 2.6835360E+001, 2.7661684E+001, 2.2733171E+001, 2.2838958E+001, 2.4403445E+001, 2.6102103E+001, 2.6622047E+001, 2.1643860E+001, 2.3007184E+001, 2.4904700E+001, 2.7019225E+001, 2.8090963E+001, 2.2196985E+001, 2.3836006E+001, 2.4572352E+001, 3.1731650E+001, 3.6554905E+001, 2.4009703E+001, 2.3513583E+001, 2.4933204E+001, 3.2237780E+001, 3.3624578E+001, 7.8555228E-001, 2.5000000E+000, 2.5000000E+000, 2.5000000E+000, 1.1289940E+000,

	9.4745239E-001, 2.8958917E-008, 5.6051997E-008, 3.1356496E-007, 8.3801231E-008, 2.9908016E-008, 4.2441263E-008, 7.8351029E-008, 1.5166648E-007, 8.4844216E-007, 2.2674909E-007, 8.0924166E-008, 1.1483601E-007, 1.8638600E-011, 3.7227289E-011, 2.0898705E-010, 5.5556247E-011, 1.9296210E-011, 2.7629477E-011, 2.0000000E+001, 2.0000000E+001, 2.7418645E+001, 3.3580323E+001, 2.0000000E+001, 2.0000000E+001
--	--

구분	실제 데이터
단위 데이터	s, kW, kW, kW, kW, kW, kW, kW, kW, kW, kg/s, kg/s
컬럼 데이터	Time, HRR, Q_RADI, Q_CONV, Q_COND, Q_DIFF, Q_PRES, Q_PART, Q_GEOM, Q_ENTH, Q _TOTAL, MLR_FUEL, MLR_TOTAL
해석값	0.0000000E+000, 0.0000000E+000, -6.1189002E-005, -1.5753424E-007, 0.0000000E+000, 0.0000000E+000, 0.0000000E+000, 0.0000000E+000, 0.0000000E+000, 0.0000000E+000, -6.1346536E-005, 0.0000000E+000, 0.0000000E+000 1.0786388E+000, 2.5084327E+000, -1.9759483E-002, 4.1883430E-002, -2.3410720E-002, 1.4457789E-004, 0.0000000E+000, 0.0000000E+000, 0.0000000E+000, 1.5064917E+000, 2.5072905E+000, 5.4754719E-005, 5.4791169E-005 ...중략... 3.5990806E+003, 0.0000000E+000, 2.7787667E-002, -2.2545588E+000, 2.0744743E+000, 1.5630302E-004, 0.0000000E+000, 0.0000000E+000, 0.0000000E+000, -1.5419669E-001, -1.5214054E-001, 0.0000000E+000, 5.0724494E-006 3.6000000E+003, 0.0000000E+000, 2.7774675E-002, -2.2439686E+000, 2.0740375E+000, 1.5610142E-004, 0.0000000E+000, 0.0000000E+000, 0.0000000E+000, -1.4200158E-001, -1.4200032E-001, 0.0000000E+000, 5.0669609E-006

- 침수 분석 데이터

침수 데이터는 plt 형식 파일이고 측정시간(현재 파일 기준 0.5초) 단위로 파일이 구분된다. 각 파일에는 컬럼 정보, 메타 데이터로 구성되어 있다. 실제 해석 데이터를 분석한 결과는 다음 표와 같다.

구분	실제 데이터
컬럼 데이터	VARIABLES = "X" "Y" "ZC" "WSL" "H" "U" "V" "W" "Q" "MAXIMUM FLOOD DEPTH"
메타 데이터	ZONE T= " 0HR 0MIN 0.5SEC", DATAPACKING=BLOCK, VARLOCATION=([3-10]=CELLCENTERED), NODES = 205, ELEMENTS = 320, ZONETYPE=FETriangle
해석값	0.00000000 0.30000000 0.60000000 0.90000000 1.20000000 1.50000000 1.80000000 2.10000000 2.40000000 2.70000000 ...중략... 155 196 195 156 197 196 157 198 197 158 199 198 159 200 199 160 201 200 161 202 201 162 203 202 163 204 203 164 205 204

(5) 데이터베이스 설계

데이터베이스는 기본적으로 메타 데이터를 저장할 수 있는 테이블과 컬럼 정보 또는 해석 정보를 저장할 수 있는 테이블을 고정형으로 구성하였으며, 컬럼 정보는 대상건물 정보에 따라 다를 수 있는 점을 감안하였고, 해석 값을 저장하는 테이블은 실시간으로 생성하도록 동적 테이블로 구성하였다.

① 층간변위 메타 데이터 및 컬럼 정보 테이블

[층간변위 메타 데이터 저장 테이블 - t_sd]

	필드명	데이터형	설명
1	id	integer	식별자
2	title	character varying(100)	제목
3	sub_title	character varying(100)	부제목
4	data_table_nm	character varying(50)	데이터 테이블명
5	structure	character varying(100)	대상건물 정보
6	analysis_series	character varying(100)	입력하중 정보
7	load_case	character varying(100)	하중조합

[층간변위 컬럼 정보 저장 테이블 - t_sd_col]

	필드명	데이터형	설명
1	id	integer	식별자
2	col_no	integer	일련번호
3	col_nm	character varying(100)	컬럼명
4	sd_id	integer	메타데이터 식별자

② 층전단력 메타 데이터 및 컬럼 정보 테이블

[층전단력 메타 데이터 저장 테이블 - t_ss]

	필드명	데이터형	설명
1	id	integer	식별자
2	title	character varying(100)	제목
3	sub_title	character varying(100)	부제목
4	data_table_nm	character varying(50)	데이터 테이블명
5	structure	character varying(100)	대상건물 정보
6	analysis_series	character varying(100)	입력하중 정보
7	load_case	character varying(100)	하중조합
8	units	character varying(20)	단위

[충전단력 컬럼 정보 저장 테이블 - t_ss_col]

	필드명	데이터형	설명
1	id	integer	식별자
2	col_no	integer	일련번호
3	col_nm	character varying(100)	컬럼명
4	ss_id	integer	메타데이터 식별자

③ 보 메타 데이터 및 해석값 저장 테이블

[보 메타 데이터 저장 테이블 - t_rg]

	필드명	데이터형	설명
1	id	integer	식별자
2	title	character varying(100)	제목
3	sub_title	character varying(100)	부제목
4	structure	character varying(100)	대상건물 정보
5	analysis_series	character varying(100)	입력하중 정보
6	load_case	character varying(100)	하중조합

[보 해석 값 저장 테이블 - t_rg_col]

	필드명	데이터형	설명
1	id	integer	식별자
2	time	numeric(14, 10)	시간
3	val	numeric(14, 10)	값
4	rg_id	integer	메타데이터 식별자

④ 기둥 메타 데이터 및 해석값 저장 테이블

[기둥 메타 데이터 저장 테이블 - t_pillar]

	필드명	데이터형	설명
1	id	integer	식별자
2	title	character varying(100)	제목
3	sub_title	character varying(100)	부제목
4	structure	character varying(100)	대상건물 정보
5	analysis_series	character varying(100)	입력하중 정보
6	load_case	character varying(100)	하중조합

[기동 해석 값 저장 테이블 - t_pillar_col]

	필드명	데이터형	설명
1	id	integer	식별자
2	time	numeric(14, 10)	시간
3	val	numeric(14, 10)	값
4	pillar_id	integer	메타데이터 식별자

⑤ 벽체 메타 데이터 및 컬럼 정보 테이블

[벽체 메타 데이터 저장 테이블 - t_w1]

	필드명	데이터형	설명
1	id	integer	식별자
2	title	character varying(100)	제목
3	sub_title	character varying(100)	부제목
4	data_table_nm	character varying(50)	데이터 테이블명
5	structure	character varying(100)	대상건물 정보
6	analysis_series	character varying(100)	입력하중 정보
7	load_case	character varying(100)	하중조합
8	units	character varying(20)	단위

[벽체 컬럼 정보 저장 테이블 - t_w1_col]

	필드명	데이터형	설명
1	id	integer	식별자
2	col_no	integer	일련번호
3	col_nm	character varying(100)	컬럼명
4	w1_id	integer	메타데이터 식별자

⑥ 화재 메타 데이터 및 컬럼 정보 테이블

[화재 메타 데이터 저장 테이블 - t_fire]

	필드명	데이터형	설명
1	id	integer	식별자
2	mw	numeric(3, 2)	MW
3	data_type	character varying(10)	devc, hrr 구분
4	data_table_nm	character varying(50)	데이터 테이블명

[화재 컬럼 정보 저장 테이블 - t_fire_col]

	필드명	데이터형	설명
1	id	integer	식별자
2	col_no	integer	일련번호
3	col_nm	character varying(100)	컬럼명
4	unit	character varying(10)	단위
5	fire_id	integer	메타데이터 식별자

⑦ 침수 메타 데이터 및 컬럼 정보 테이블

[침수 메타 데이터 저장 테이블 - t_flood]

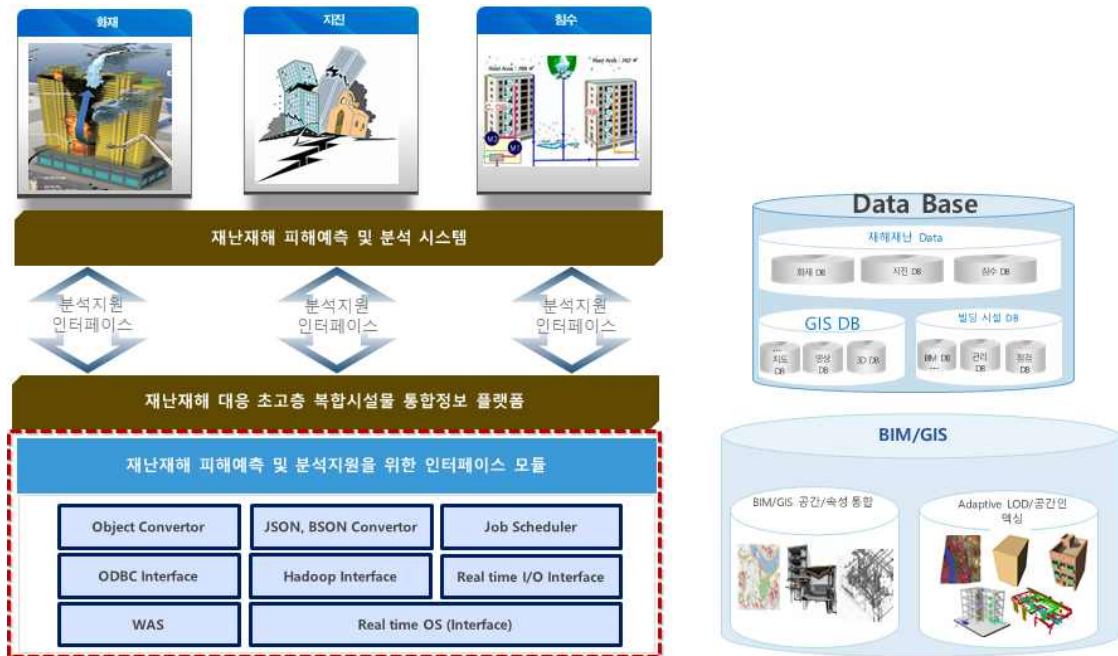
	필드명	데이터형	설명
1	id	integer	식별자
2	time_hr	integer	시간(시)
3	time_min	integer	시간(분)
4	time_sec	numeric(3, 1)	시간(초)
5	datapacking	varchar(50)	DATAPACKING
6	varlocation	varchar(50)	VARLOCATION
7	nodes	varchar(50)	NODES
8	elements	varchar(50)	ELEMENTS
9	zonetype	varchar(50)	ZONETYPE

[침수 해석 값 저장 테이블 - t_flood]

	필드명	데이터형	설명
1	id	integer	식별자
2	x	numeric(10, 4)	X
3	y	numeric(10, 4)	Y
4	zc	numeric(10, 4)	ZC
5	wsl	numeric(10, 4)	WSL
6	h	numeric(10, 4)	H
7	flood_id	integer	메타데이터 식별자

(6) 플랫폼과 연계할 수 있는 인터페이스 모듈 개발

플랫폼과 연계하는 인터페이스 모듈의 구조도는 다음 그림과 같다.

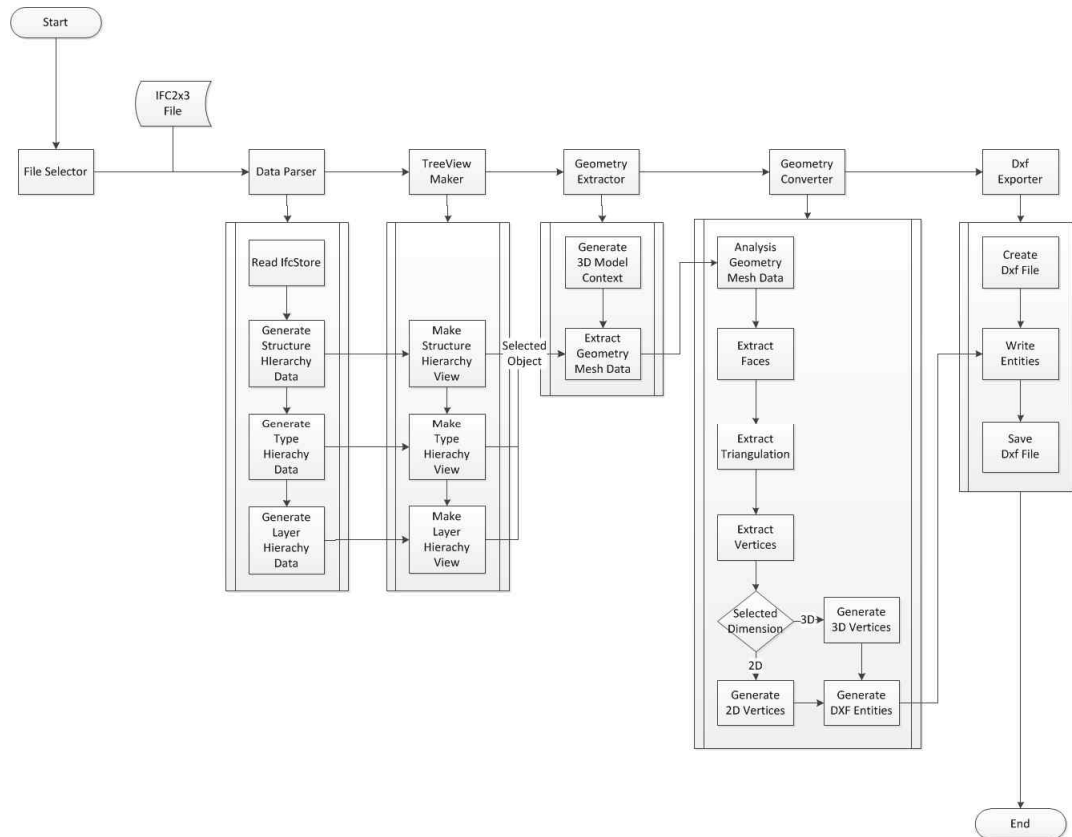


[인터페이스 모듈 구조도]

① Export 모듈 구현

Export 모듈의 동작 절차는 데이터의 구문 분석, 트리 구조 시각화, 공간객체 추출, 공간객체 변환, 도면파일 저장 단계를 거치도록 구현하였다. 우선적으로 사용자가 변환 대상파일 (IFC2)을 선택하면 해당 파일을 읽어 변환 대상 데이터의 구문 분석을 수행한다. 해당 절차에서는 건물의 구조 정보에 대한 계층 정보와 각 객체의 유형에 대한 계층 정보, 각 레이어에 대한 계층 정보를 추출한다. 이렇게 추출한 각 계층별 데이터를 사용자에게 트리 구조로 표현하고 해당 트리 구조에서 사용자가 Export하고자 하는 객체를 선택하면, 변환 대상파일에서 선택한 객체의 공간객체(Geometry)를 추출한다. 추출된 공간객체를 분석하여 공간객체를 표현할 수 있는 면(Face)을 추출하고 해당 면을 구성하고 있는 기본 단위인 삼각객체(Triangulation)을 추출한 후 해당 객체의 정점(Vertices)을 추출한다. 이렇게 추출한 데이터를 3D로 Export할 경우에는 3D 정점을 생성하고, 2D로 Export할 경우에는 2D 정점을 생성하여 도면(DXF)파일에서 사용하는 객체(Entity)로 재구성한다. 재구성된 객체를 활용하여 최종적으로 도면파일을 생성한다.

2) Industry Foundation Classes의 약자로 building SMART(구 IAI)가 개발한 건물의 3차원 형상 및 BIM 데이터에 내포된 속성 정보를 서로 다른 소프트웨어 사이에서 주고받을 수 있는 포맷.



[Export 모듈의 데이터 처리 절차 플로우차트]

② Export 사용자 인터페이스

각 재난유형의 [Export] 버튼을 클릭하면 변환 대상파일을 선택할 수 있는 다이얼로그가 생성된다. 변환 대상파일 선택 다이얼로그에서 변환 대상파일을 선택하면 해당 파일에 존재하는 건물정보를 해석하여 각 계층별 데이터를 트리구조로 조회할 수 있다. 해당 트리구조에서 변환하고자 하는 객체를 선택하고, 원하는 차원(2D or 3D)에 해당하는 버튼을 클릭하면 파일 저장 다이얼로그가 생성된다. 해당 다이얼로그에 저장할 파일명을 입력하면 사용자가 지정한 폴더에 도면파일이 생성된다. 다음 그림은 해당 절차를 표현한 그림이다.

The diagram illustrates the relationship between IFC product and shape representations. It shows a hierarchy starting from an **(ABS) IfcProduct** (e.g., IfcWall), which has a **Representation (INV) ShapeOfProduct**. This leads to an **IfcProductDefinitionShape** (subtype of IfcProductRepresentation), which in turn has **Representations (INV) OfProductRepresentation**. These are represented by **IfcShapeRepresentation** (subtype of IfcRepresentation). The **IfcShapeRepresentation** is linked to **(ABS) IfcGeometric RepresentationItem** (e.g., IfcExtrudedAreaSolid) via **Items[1]** and **Items[2]**. These items are then assigned to **IfcPresentationLayer Assignment** objects via **AssignedItems (INV) LayerAssignment**. The diagram shows two assignments: one for **A_COL_01** and another for **A_COL_01-A**. A 3D model of a column is shown, with a red box highlighting the **IfcPresentationLayer Assignment** for **A_COL_01-A**. A note indicates that the assignment to **IfcShapeRepresentation** is overridden by the assignment to **IfcGeometricRepresentationItem**.

The screenshot shows the **Export 대상 선택** (Export Selection) dialog. The **Layers** tab is selected, showing a list of layers with their corresponding IFC representations. The layers are:

- [Layer] A-AREA-----OTLN
- [Layer] A-CLING-----OTLN
- [Layer] A-DOOR-----OTLN
- [Layer] A-FLOOR-----OTLN
- [Layer] A-FLOOR-HRAL-OTLN
- [Layer] A-GLAZ-----OTLN
- [Layer] A-GLAZ-CURT-OTLN
- [Layer] A-GLAZ-CWMS-OTLN
- [Layer] A-MASS-----OTLN
- [Layer] A-NPLT-----OTLN
- [Layer] A-ROOF-----OTLN
- [Layer] A-WALL-----OTLN
- [Layer] Q-SPCO-----OTLN
- [Layer] S-BEAM-----OTLN
- [Layer] S-COLS-----OTLN
- [Layer] S-FNDN-----OTLN
- [Layer] S-STRS-----OTLN

Buttons for **Export 3D** and **Export 2D** are visible at the bottom.

- 105 -

6. 실시간 재난 정보의 3D 시설물 표출 시스템 시제품 제작

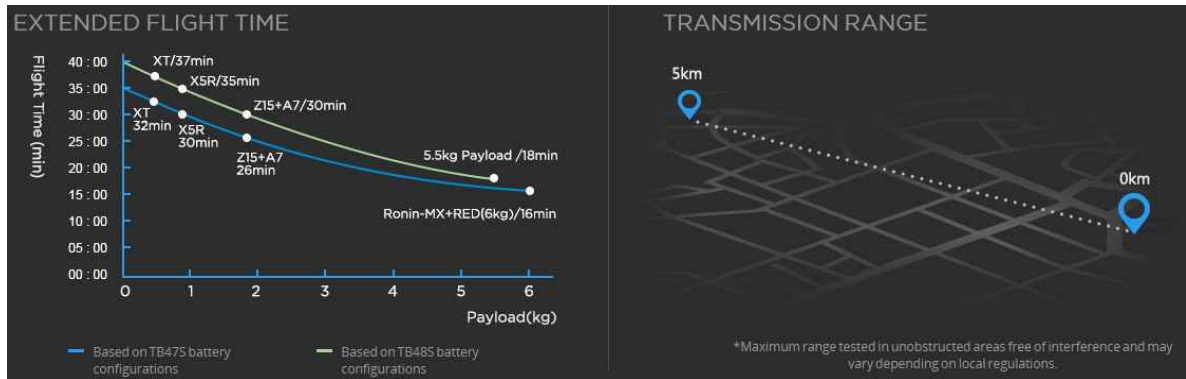
(1) 시설물 촬영을 위한 드론과 스캐닝 센서

○ 스캐닝 센싱이 가능한 드론 사양

- 드론에 스캐닝 센서를 탑재하여 지상 쪽 방향을 촬영 할 수 있는 드론을 제작하였다. 최대한 가벼운 사양의 센서를 탑재하여 기동시간을 최대화하였고 배터리 모듈 등을 최적화하였다.
- 드론은 풍속 10m/s의 바람에서 비행이 가능하고, 프로펠러 1개 정지 시에도 잔여 프로펠러로 안전하게 비행 및 착륙이 가능하다.
- 비행속도 25km/h 이상, 비행시간 1회 20분 이상, 배터리를 제외한 총 중량은 12kg 이하이며 충돌방지를 위한 안전장치 기능이 있다.

○ 비행 드론의 구성

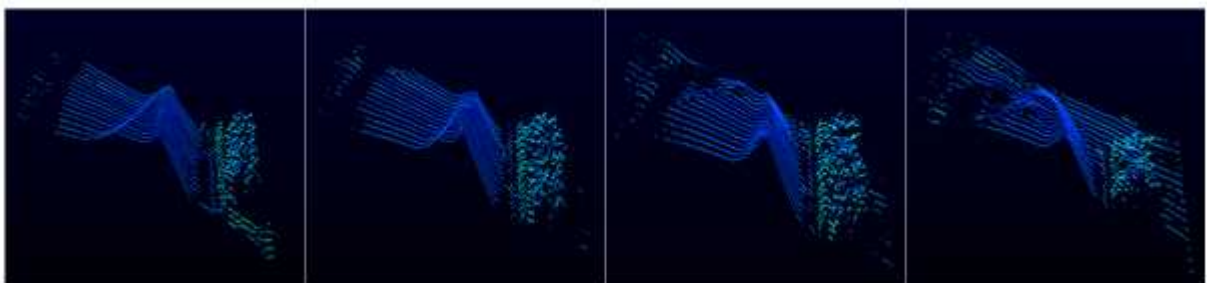
- 시설물의 촬영을 위한 고도 기반의 스캐닝 센서 장착의 일반적인 기본 비행 드론 사양
 - 프로펠러 : Hhxa-Copter형 6개
 - 본 기체 및 촬영장비포함 총 중량 10kg 이하(배터리 제외)
 - 비행 컨트롤시스템(F/C) : A3 Pro
 - 최대 바람저항 : 8m/s
 - 최대속도 : 무풍시 657km/h
 - 작동온도 : -10℃ ~ 40℃
 - 조종기 작동 주파수 : 2.400GHz~2.483GHz
 - 최대 전송 거리 : 5km
- 비행 컨트롤 시스템(F/C : Flying Controller) 인 A3 Pro는 2채널 전송 시스템, 지능형 배터리 및 배터리 관리 시스템을 포함 하고 있다.
- 최대 송수신 모듈은 2.4Ghz 통신 주파수 안테나장치로 구성 되어 있고 최대 송수신 거리는 5Km까지 가능하다.



[무게에 따른 비행 시간과 설정 할 수 있는 최대 송수신거리]

○ 라이다 측정기 사양

- 360도 스캐닝 기능을 가지고 있는 라이다 센서
 - 기본적으로 라이다는 360도 회전을 하면서 주변 환경을 스캐닝한다.
 - 라이다를 드론에 장착하여 상공에서 지상을 촬영 및 스캐닝 할 경우 지상방향으로 필요한 약 90도 정도만 한다.
 - 라이더의 세팅기능을 이용하여 촬영각을 조정하거나 저장된 데이터에서 필요한 각의 범위 내의 값만 추출하여 사용한다.
- 측정범위 40미터 초당 1,000 반사데이터 수신 가능
 - 측정범위 40미터이므로 신뢰성 있는 데이터의 습득을 위해 고도 약 25~30미터 이내로 촬영하였다.
 - 초당 1000개의 빔을 발사하여 수신이 가능하지만 360도의 1/4인 약 90도만 필요하므로 초당 약 250개의 빔을 해석하여 사용 가능하다.
- single beam 스캐너로 single plane으로 데이터 기록
 - 빔을 발사 할 때 1개채널로 빔을 발사하여 데이터의 3D이미지화가 명료하다.
 - 16채널 및 32채널로 발사하여 반사데이터를 수신할 경우 이전프레임과 현재프레임의 반사의 중첩점이 많아져 3D이미지화 할 때 중첩점의 필터링이 까다로워진다.



[각 프레임의 반사점 중첩]

- 반사 Raw데이터는 저장하여 PC로 송신가능
- 라이다는 반사된 데이터를 USB 케이블등을 통해서 PC로 송신이 가능하다.
- Raw 데이터는 텍스트 형식의 데이터로서 가공 및 이용이 용이하다.

```
{ "bSensorIsMobile": false, "bLogTimeStampPerSweep": true, "Sweeps": [ { "TimeStamp": 1479764674203, "SensorReading_Angles": [ 270.5, 271.1875, 271.875, 272.5, 273.1875, 273.875, 274.5625, 275.1875, 275.875, 276.5625, 277.1875, 277.875, 278.5625, 279.1875, 279.875, 280.5625, 281.25, 281.875, 282.5625, 283.25, 283.875, 284.5625, 285.25, 285.875, 286.5, 287.1875, 287.8125, 288.5, 289.1875, 289.8125, 290.5, 291.1875, 291.8125, 292.5, 293.1875, 293.8125, 294.5, 295.1875, 295.8125, 296.5, 297.1875, 297.8125, 298.5, 299.1875, 299.8125, 300.5, 301.1875, 301.8125, 302.5, 303.1875, 303.8125, 304.5, 305.1875, 305.8125, 306.5, 307.1875, 307.8125, 308.4375, 309.125, 309.75, 310.4375, 311.0625, 311.75, 312.4375, 313.0625, 313.75, 314.4375, 315.0625, 315.75, 316.4375, 317.0625, 317.75, 318.375, 319.0625, 319.75, 320.375, 321.0625, 321.75, 322
```

[라이다의 Raw 데이터]

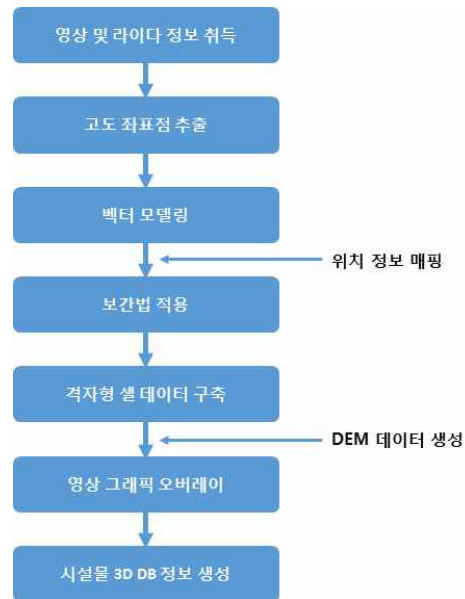


[라이다 및 USB 연결 케이블]

(2) 스캐닝 촬영 데이터를 이용한 가상 모형 구축

○ 3D 시설물 DB 생성 프로세스

- 3D 시설물 DB 생성 프로세스는 다음과 같다.

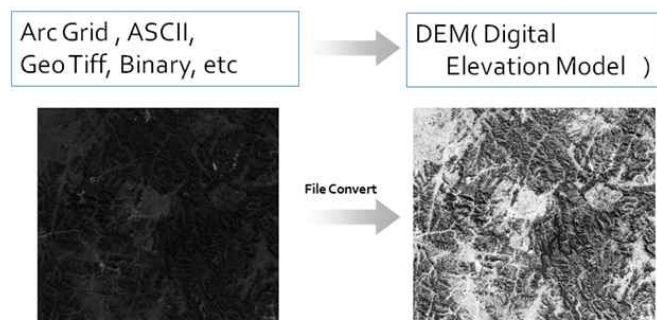


[시설물 3D DB 생성 프로세스]

- 경위도의 2D 정보를 갖는 표면 정보를 3D의 공간정보로 표현하기 위해 Z축, 즉 높이값 필요하므로 드론이 측정한 고도값으로 수치표고모형(DEM : Digital Elevation Model)을 생성하고 이를 영상 정보에 오버레이하여 최종 결과물을 생성한다.

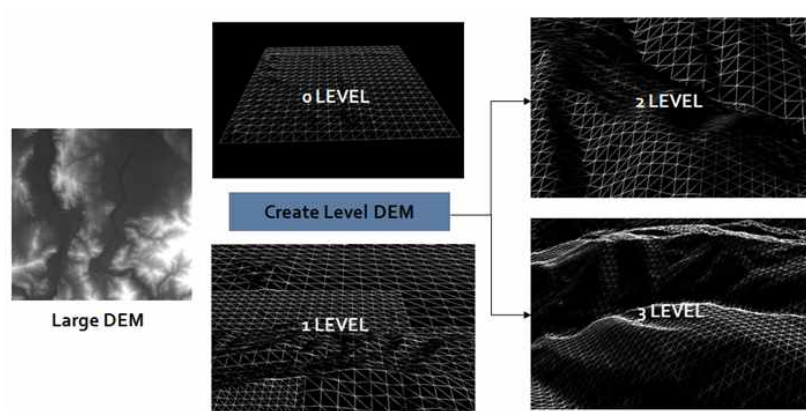
○ 3차원 데이터 변환 및 모델링 절차

- 절차 1. DEM 전처리 과정 즉, 여러 형태의 지형지물 파일을 계층 구조로 변환하기 위해 프로그램에서 수용 가능한 통합 파일 형태의 임시 파일로 저장



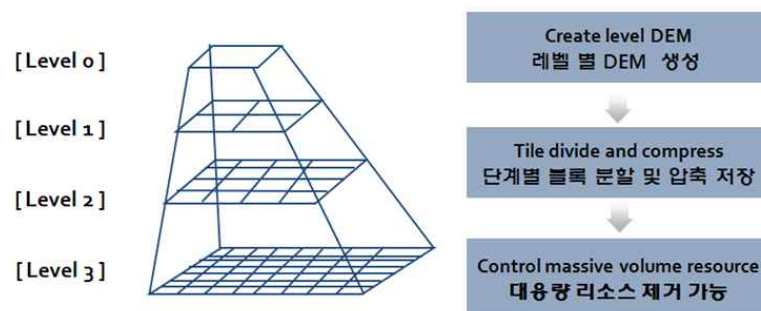
[DEM 파일 저장]

- 절차 2. DEM의 resolution을 부모자식 상관관계를 위해 2배수로 조절하여 각 레벨을 생성



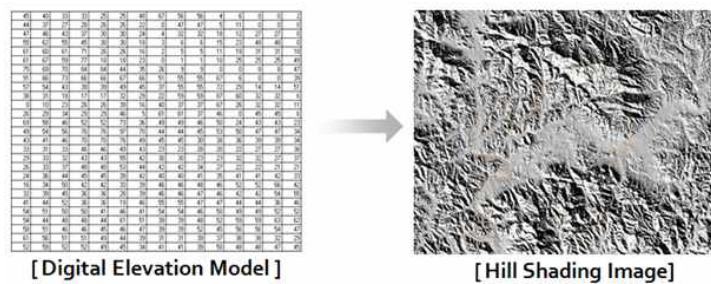
[Level DEM 생성]

- 절차 3. 각 단계 DEM을 2의 자승 크기(32 or 64)의 정사각형 블록 형태로 분할하여 저장



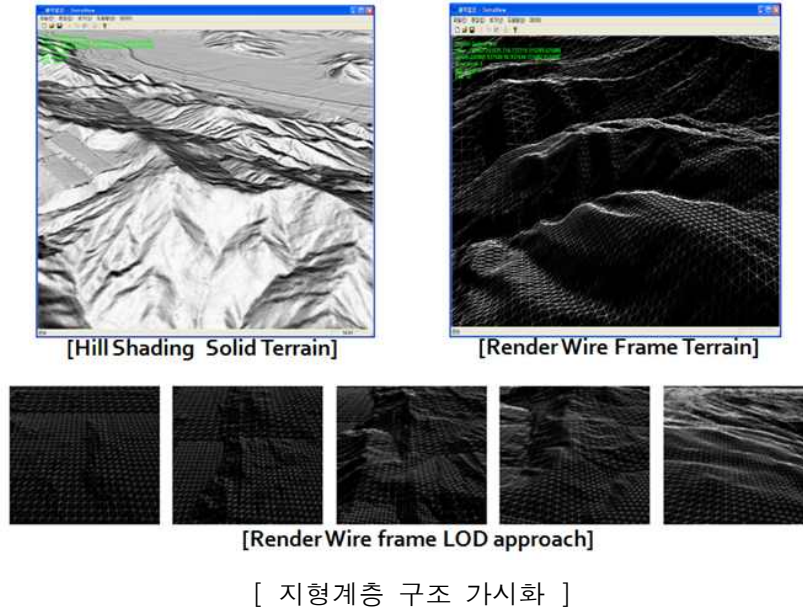
[단계별 지형 분할]

- 절차 4. 각 단계별로 분할된 지형의 계층 구조를 생성한다. 지형지물 블록의 음영기복 이미지를 생성



[음영기복 이미지 생성]

- 절차 5. 지형계층 구조를 가시화 작업 구현

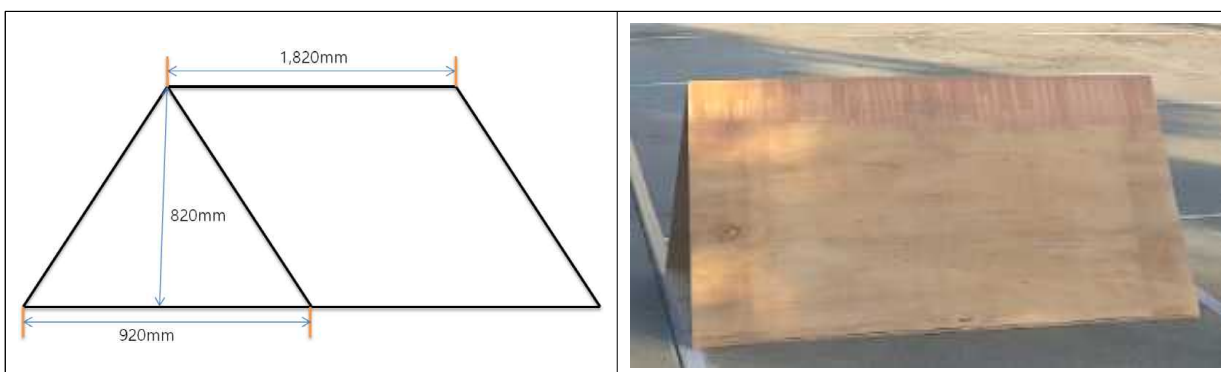


- 벡터 기반의 2차원 화면 출력 방식에서 3차원 데이터를 각종 지형물에 올리기 위해서는 아래의 사항이 충족되어야 한다.

- 지형지물의 원래 데이터 내용 및 의미를 변경하지 않아야 함
- 3차원 데이터를 출력할 수 있도록 설계되어야 함
- 범용 가능한 3차원 출력이 가능하도록 설계되어야 함

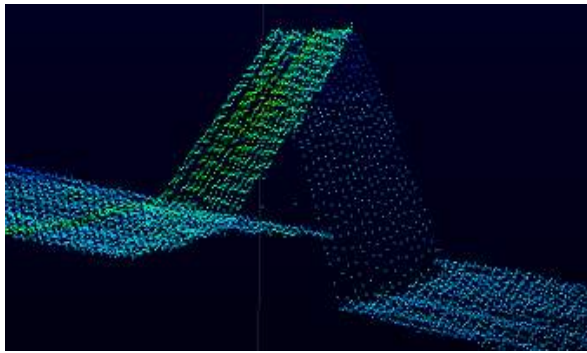
○ 가상 모형물 구축

- 드론 스캐닝 센서의 성능 시험을 위해 삼각 기둥 모형을 제작 하였으며 시험 촬영을 하여 데이터를 획득 하였다.



[3D 데이터 획득을 위한 모형 제작]

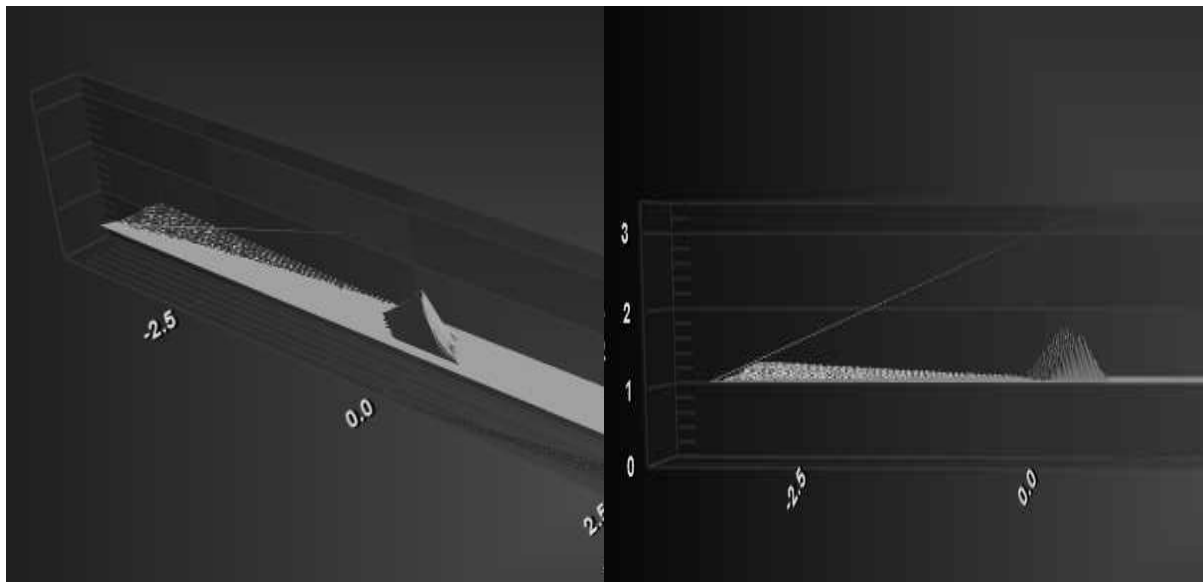
- 실외 고도 3m에서 드론으로 촬영하여 스캐닝 센서 장비회사가 제공하는 툴로 모델링하였다. 이 툴은 단순히 3D 데이터를 분석하여 화면에 모니터링을 할 수 있는 기능과 3D 데이터를 일반적인 포맷의 파일로 변환하는 기능을 가지고 있다.



[3D 모델링]

○ 촬영 데이터를 이용한 대상 모델링 순서

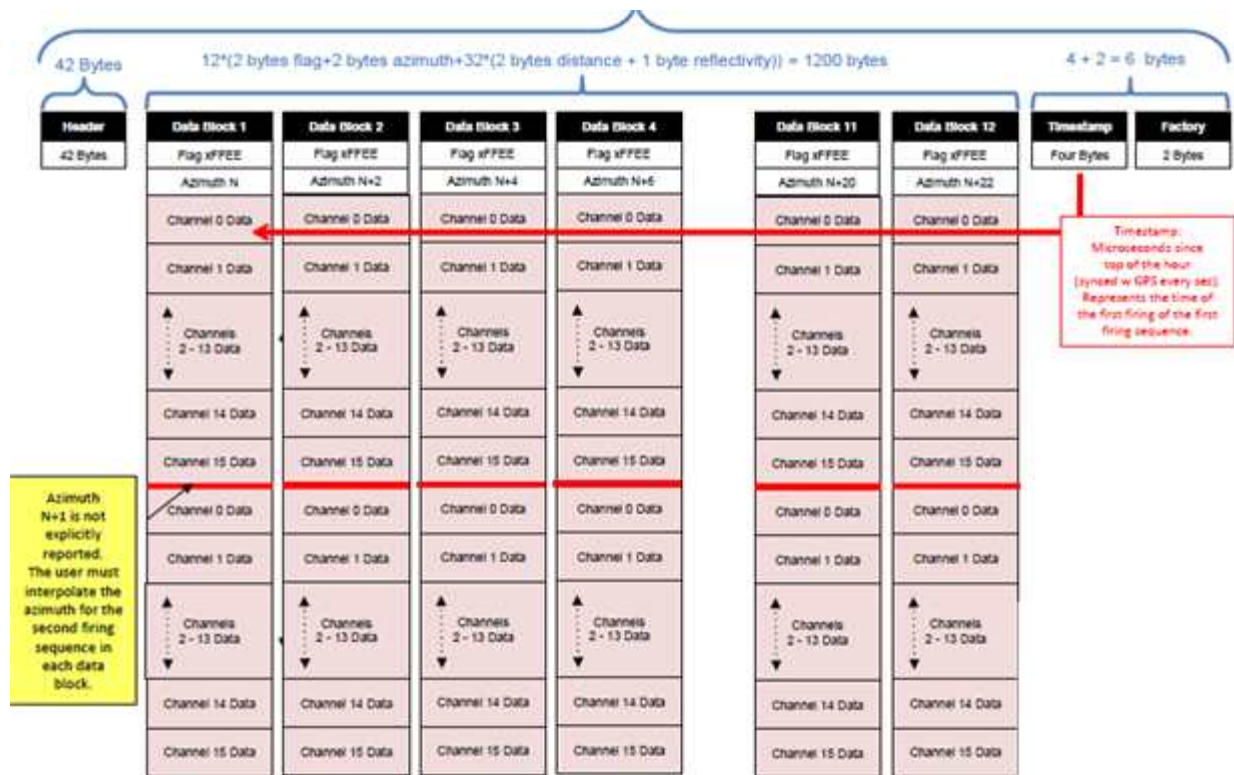
- 드론으로 촬영한 스캐닝 데이터는 바이너리 데이터로서 촬영 후 드론으로부터 모델링 시스템으로 복사한다.
- 스캐닝 센서 장비회사가 제공하는 3D 변환 툴을 이용하여 모델링 시스템이 분석 할 수 있는 데이터로 변환 한다.
- 변환된 데이터를 모델링 시스템의 특정 폴더로 복사 한다.
- 모델링 시스템은 변환된 데이터를 읽어 편집 및 수정이 가능한 형태로 화면에 모델링 한다.



[삼각기둥의 3D 표출]

○ 스캐닝 데이터의 구조

- 스캐닝 센서로 촬영한 데이터는 pcap 형식의 데이터로 위치데이터 XYZ, 방위각, 피사체와의 거리, 반사강도, 레이저 아이디, 타임스탬프의 정보를 저장하고 있다.



[pcap 바이너리 데이터]

- 스캐닝 센서의 촬영 단위는 프레임이며 각 프레임에는 센서가 한번 촬영한 약 8000여개의 빔이 포함 되어 있고 360도의 회전을 통해 사방의 반사데이터를 저장한다. 스캐닝 센서는 촬영 방향을 전후좌우 4곳중 한방향만의 데이터만 필요 하므로 레이저 발사 각도는 90도로 설정하여 앞쪽에 위치한 시설물만 촬영하게 된다. 위치데이터인 XYZ에서 X는 폭의 데이터를 나타내고 Y는 높이 데이터, Z는 진행방향의 데이터를 표현한다. pcap 데이터는 여러개의 프레임단위로 장되어 있으며 스캐닝 센서 장비회사가 제공하는 3D 툴에서 pcap 데이터는 프레임단위로 화면에 출력한다.

○ 프레임 추출 및 저장

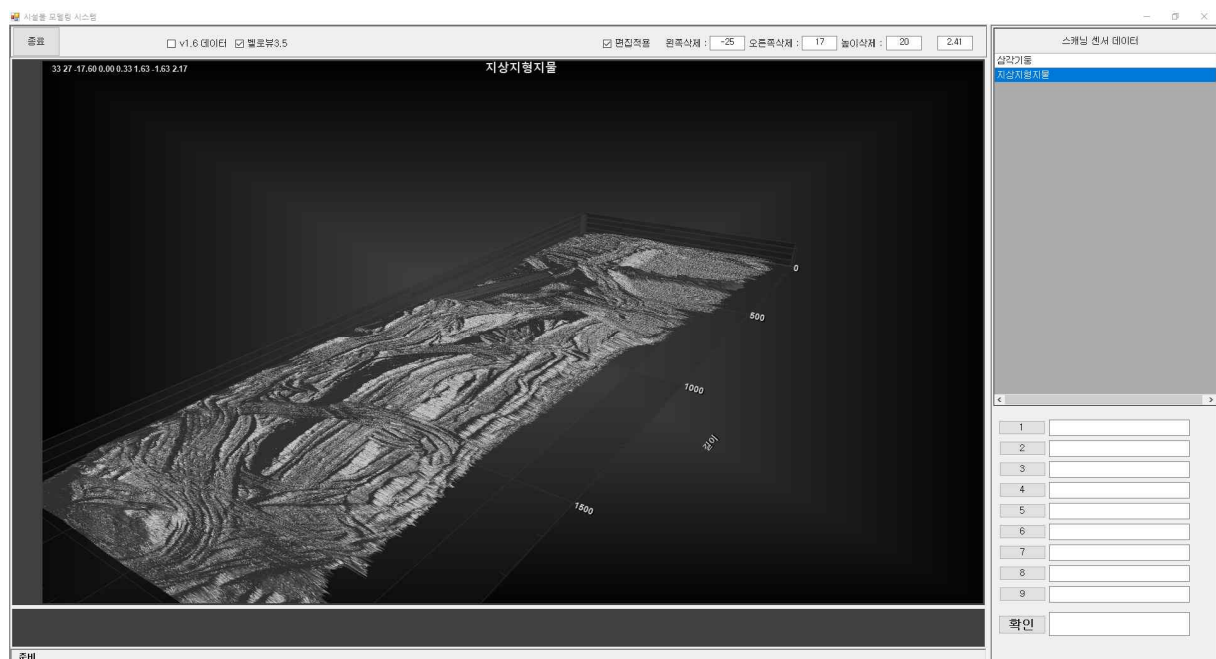
- pcap 데이터는 한 화면에 한 프레임을 출력하므로 화면에 출력하기 위해서는 여러 프레임을 한 데이터 묶음으로 추출하여 한다. 이 작업은 시작되는 프레임부터 끝나는 프레임까지 육안을 통해 지정하며 정확도에 따라 모델링 구축에 영향을 주게 된다. 즉 데이터의 처음과 끝이 표현된 프레임을 정확히 별도 파일(CSV)로 저장하여야 한다. CSV 파일은 프레임 단위로 각각 저장 된다.

Points_m_	Points_m_	Points_m_	intensity	laser_id	azimuth	distance_m	adjustedtime	timestamp
0.053809	34.25555	-9.17876	12	0	9	35.464	258906888	2.59E+08
0.054781	31.38717	0.547866	7	1	10	31.392	258906890	2.59E+08
0.065301	34.01325	-7.85259	11	2	11	34.908	258906892	2.59E+08

[CSV 파일 구조]

○ CSV파일의 합병

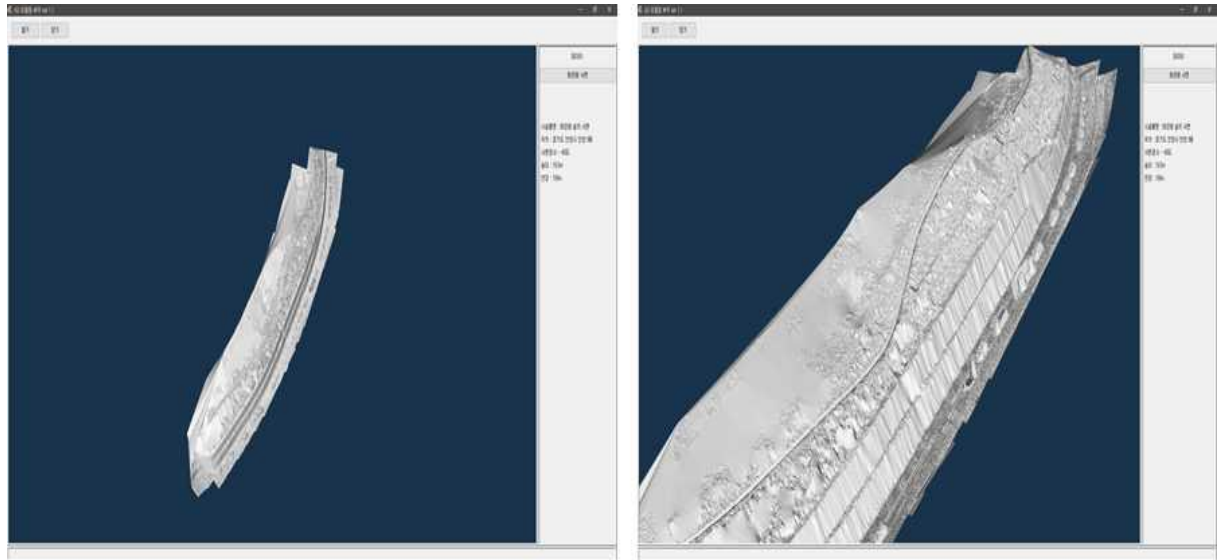
- 프레임 단위로 저장된 CSV파일에서 시설물의 폭과 높이는 3D 모델링으로는 정확한 값이지만 진행방향이 Z는 프레임마다 거의 같은 값을 가지고 있어 CSV 파일을 모두 합병하였을 경우 모두 중첩되어 이 경우 모델링에 많은 오차 값이 발생 하게 된다. 그러므로 각 프레임 단위 파일에서 가장 중앙에 위치한 데이터를 추출하고 빔과 빔사이의 거리를 계산하여 Z방향으로 증가하여 저장한다.



[지상 지형지물 3D 모델링]

○ VTK를 이용한 3D 시설물 표출 시스템 시제품 개발

- 고도 데이터가 있는 자동 도면을 가공하여 상하좌우, 줌인/줌아웃, 회전등이 가능하여 현장의 형상을 가늠할 수 있는 3D 시설물 표출 시스템(시제품)을 개발 하였다.



[3D 시설물 표출 시스템]

제4장. 목표 달성도 및 관련 분야 기여도

1. 목표 달성도

정확하고 효율적인 시설물 유지 관리를 위해 현장용 계측 장비와 연계하여 시설물 내 입체적 위치관리가 가능한 3차원 시설물 모형의 자동 구축을 가능하게 지원하는 SW 엔진 모듈과 재난 재해에 대비한 유지 관리용 감시 시스템에 IoT 기술을 접목하여 스마트폰과 연동이 가능한 인터페이스 및 3D 시설물 관리를 위한 스마트 유지관리 통합 플랫폼 기반 기술을 성공적으로 개발하였음.

기술 환경, 경쟁자 등 제반 환경을 지속적으로 모니터링 하면서 연구 계획을 지속적으로 보완하여 최적의 성과를 도출할 수 있도록 추진하였음. 기본적으로 범용성, 저작가능성, 표준성, 생산성, 정확성, 자동화의 6대 원칙을 기반으로 협의 추진하여 성과를 도출하였음.

연구 개발진들은 기본적으로 연구 성과의 우수성, 창의성, 파급효과, 활용 가능성에 대한 강한 확신을 가지고 연구하여 계획대비 모든 실적을 100% 충족하였음.

세부 연구 목표 (연구 계획서의 목표)	비중 (%)	달성도 (%)	자체 평가
위치 기반 3D 모델링 및 시범 DB 설계 및 구축	25%	100%	우수
무선통신 기반 계측센서 및 시설물 3D 정보 연동 인터페이스 모듈 개발	25%	100%	우수
드론 기반 3D 엔진 기반 모듈 개발	25%	100%	우수
무선 제어 계측 통합 플랫폼 기반 기술 개발	25%	100%	우수
합계	100점	100%	

[총괄]

구분	핵심성과 (CTE)		단위성과	성과점검기준						양적 성과 건수
				질적 성과지표	목표치	달성치	측정방법	검증방법	가중치	
주관 대림대학교	1	위치 기반 3D 모델링 및 시범 DB 설계 및 구 축	1-1 - 위치(GPS) 및 촬영 정보를 이용한 위치 기반 3D 모델링 모듈 개발 - 스마트 유지관리 기반을 위한 시설물 3D 속성정보와 연계된 시범 DB 설계 및 구축	① 3D 모델링 알고리즘 및 시범 DB 설계서	DB 설계서	100%	DB 설계서 확인	DB 설계서 확인	20	특허: 출원 1건 등록 1건 논문: 1건 시제품: 1건 S/W: 1건
	2	무선통신 기반 계측센서 및 시설물 3D 정보 연동 인터페이스 모듈 개발	2-1 - 현장용 무선통신 기반 계측센서 다중 인터페이스 연동 모듈개발 - 수집된 계측 정보의 정확한 입체적 관리를 위한 시설물 3D 정보 연동 인터페이스 기술 구현	① 기능구현 정확성	최소 4종류 이상의 센서를 연동	100%	계측기 연동 테스트	계측기 연동 실시간 정보 확인	30	특허: 출원 1건 등록 1건 논문: 2건 시제품: 1건 S/W: 1건
공동 (주)케이시아이	3	드론 기반 3D 엔진 기반 모듈 개발	3-1 - 라이다 카메라를 장착한 드론을 이용하여 대상 시설물 촬영 및 영상정보 수집 - 시설물 내 센서 정보의 입체적 위치관리가 가능한 시설물 3D 가상 모형 구축 모듈 개발 - 계측 정보 데이터베이스와 연결이 가능한 3D 시설물 뷰어 모듈 개발	① 기능구현 정확성	최소 3종의 시설물에 대한 3D 모델링 수행	100%	프로그램 테스트	뷰어 프로그램을 통한 시설물 확인	20	특허: 출원 1건 등록 1건 논문: 1건 시제품: 1건 S/W: 1건
	4	무선 제어 통합 플랫폼 기반 기술 개발	4-1 - 무선 네트워크를 이용하여 계측기와 연동이 가능한 스마트폰용 통신 관리 앱 모듈 개발 - 재난재해(지진 및 수재해 등)용 통합 정보 플랫폼과의 효과적인 연계를 위한 (시설물 및 감시 정보) 인터페이스 모듈 구현 - 실시간 재난 정보의 3D 시설물 표출 시스템 시제품 제작	① 기능구현 정확성	4종류 이상의 센서와 3종 이상의 시설물과 연동한 통합 플랫폼 구축	100%	프로그램 테스트	시제품 시연	30	특허: 출원 1건 등록 1건 논문: 1건 시제품: 1건 S/W: 1건
계									100	특허: 출원 2건 등록 2건 논문: 2건 시제품: 2건 S/W: 2건

○ 핵심 연구성과

연번	핵심성과 (CTE)	핵심성과 설명	성과유형 ¹⁾	연구기관	해당 연차	TRL
1	위치 기반 3D 모델링 및 시범 DB 설계 및 구축	○ 위치(GPS) 및 촬영 정보를 이용한 위치 기반 3D 모델링 모듈 개발 ○ 스마트 유지관리 기반을 위한 시설물 3D 속성정보와 연계된 시범 DB 설계 및 구축	소프트웨어	대림대학교 산학협력단	1년차	4
2	드론 기반 3D 엔진 기반 모듈 개발	○ 라이다 카메라를 장착한 드론을 이용하여 대상 시설물 촬영 및 영상정보 수집 ○ 시설물 내 센서 정보의 입체적 위치관리가 가능한 시설물 3D 가상 모형 구축 모듈 개발 ○ 계측 정보 데이터베이스와 연결이 가능한 3D 시설물 뷰어 모듈 개발	소프트웨어	(주)케이지아이	1년차	4
3	무선통신 기반 계측센서 및 시설물 3D 정보 연동 인터페이스 모듈 개발	○ 현장용 무선통신 기반 계측센서 다중 인터페이스 연동 모듈 개발 ○ 수집된 계측 정보의 정확한 입체적 관리를 위한 시설물 3D 정보 연동 인터페이스 모듈 구현	소프트웨어	대림대학교 산학협력단	2년차	4
4	무선 제어 계측 통합 플랫폼 기반 기술 개발	○ 무선(블루투스 , WiFi/LTE) 네트워크를 이용하여 계측기와 연동이 가능한 스마트폰용 통신 관리 앱 모듈 개발 ○ 재난재해(지진 및 수재해 등)용 통합 정보 플랫폼과의 효과적인 연계를 위한 (시설물 및 감시 정보) 인터페이스 모듈 구현 ○ 실시간 재난 정보의 3D 시설물 표출 시스템 시제품 제작	시스템	(주)케이지아이	2년차	4

○ 핵심 연구성과별 설명

핵심성과 1		
정의	위치 기반 3D 모델링 및 시범 DB 설계 및 구축	
목적	■ 위치(GPS) 및 촬영 정보를 이용한 위치 기반 3D 모델링 모듈 개발 ■ 스마트 유지관리 기반을 위한 시설물 3D 속성정보와 연계된 시범 DB 설계 및 구축	
성능목표	(기존기술 수준) 기존 드론으로 수집된 정보는 2차원 영상이었으며 일부 Z값을 보유한 데이터를 통해 3D 형태의 수치표고모형을 작성함 (개발기술 수준) 드론으로부터 수집된 촬영영상과 라이다로부터 측정된 Z 값을 이용하여 3D 형태의 수치표고모형을 작성	
성과달성 책임자 (소속기관)	이영걸(대림대학교 산학협력단)	
소요기간	2017.04~2017.12(1년차)	
성과달성 확인방법	3D 모델링 알고리즘 및 시범 DB 설계서 확인	
성과달성에 따른 정량적 성과	특허 출원/등록	
	논문	1건
	저작	
	S/W	
	신기술	
	기타	

핵심성과 2		
정의	드론 기반 3D 엔진 기반 모듈 개발	
목적	■ 라이다 카메라를 장착한 드론을 이용하여 대상 시설물 촬영 및 영상정보 수집 ■ 시설물 내 센서 정보의 입체적 위치관리가 가능한 시설물 3D 가상 모형 구축 모듈 개발 ■ 계측 정보 데이터베이스와 연결이 가능한 3D 시설물 뷰어 모듈 개발	
성능목표	(기존기술 수준) 기존에는 2차원의 평면상에 시설물 정보를 단순 관리 (개발기술 수준) 3D 형태의 시설물 내 센서 정보의 입체적 위치관리가 가능하도록 개발	
성과달성 책임자 (소속기관)	이윤래((주)케이지아이)	
소요기간	2017.04~2017.12(1년차)	
성과달성 확인방법	3D 시설물 뷰어를 통한 SOC 시설물 2개, 일반 시설물 1개, 총 3개의 3D 시설물 정보 확인	
성과달성에 따른 정량적 성과	특허 출원/등록	출원 1건
	논문	
	저작	
	S/W	S/W 등록 1건
	신기술	
	기타	기술이전 1 건

핵심성과 3		
정의	무선통신 기반 계측센서 및 시설물 3D 정보 연동 인터페이스 모듈 개발	
목적	○ 현장용 무선통신 기반 계측센서 다중 인터페이스 연동 모듈 개발 ○ 수집된 계측 정보의 정확한 입체적 관리를 위한 시설물 3D 정보 연동 인터페이스 모듈 구현	
성능목표	(기존기술 수준) 유선 위주의 센서 및 계측기 연동 (개발기술 수준) 무선 위주의 실시간 센서 네트워크로 구성하고 방대한 데이터를 효율적으로 처리하기 위한 알고리즘 구현	
성과달성 책임자 (소속기관)	이영걸(대림대학교 산학협력단)	
소요기간	2018.01~2018.12(2년차)	
성과달성 확인방법	계측기로부터 실시간으로 들어오는 정보 확인	
성과달성에 따른 정량적 성과	특허 출원/등록	
	논문	2건
	저작	
	S/W	
	신기술	
	기타	

핵심성과 4		
정의	무선 제어 계측 통합 플랫폼 기반 기술 개발	
목적	○ 무선(블루투스 , WiFi/LTE) 네트워크를 이용하여 계측기와 연동이 가능한 스마트폰용 통신 관리 앱 모듈 개발 ○ 재난재해(지진 및 수재해 등)용 통합 정보 플랫폼과의 효과적인 연계를 위한 (시설물 및 감시 정보) 인터페이스 모듈 구현 ○ 실시간 재난 정보의 3D 시설물 표출 시스템 시제품 제작	
성능목표	(기존기술 수준) 유선 네트워크를 이용한 센서 기반 모니터링 시스템 위주 (개발기술 수준) 무선 스마트 센서를 통해 데이터를 송수신 및 제어가 가능하도록 앱 개발 및 전체 시스템 구성	
성과달성 책임자 (소속기관)	이윤래((주)케이지아이)	
소요기간	2018.01~2017.12(2년차)	
성과달성 확인방법	앱 및 전체 시스템에 대한 시범 시스템을 통한 구동 여부 확인	
성과달성에 따른 정량적 성과	특허 출원/등록	출원 1건
	논문	
	저작	
	S/W	1건
	신기술	
	기타	

다. 정량적 연구성과 목표 및 달성도

구분		목표	달성도(%)	비고
국외 논문 게재	SCI			
	비SCI			
	계			
국내 논문 게재	SCI			
	비SCI	2건	150%	초과달성(총3건)
	계	2건	150%	
특허 출원	국내	2건	100%	
	국외			
	계	2건	100%	
특허 등록	국내			
	국외			
	계			
인력 양성	석사			
	박사			
	계			
기술 이전	계약 건수	0건	1건	초과달성
	계약액		5,000,000원	
	징수액			
국제 표준화	기고서 제출			
	기고서 채택			
	표준안 채택			
국내 표준화	기고서 제출			
	기고서 채택			
	표준안 채택			
등록·기탁 대상 성과물	보고서 원문			
	연구시설·장비			
	기술요약정보			
	생명자원	생명정보		
		생물자원		
		신품종		
	소프트웨어		2건	100%
기타	화학물			
	시제품			
	성과 홍보			
	일자리 창출			

라. 성과 목표에 대한 자체 평가

성과 목표	자체 평가
논문 2건 특허출원 2건 소프트웨어 등록 2건	논문 3건, 특허출원 2건, 소프트웨어 등록 2건 등 당초 성과 목표 100% 달성 외에 기술 이전 계약 실시로 성과 목표 이상의 결과를 도출함

2. 관련 분야 기여도

- 드론의 카메라 및 측정 장치를 활용함으로써 실시간으로 대상 시설물의 3차원 형태와 변이 상태를 감시하고, 수집된 데이터를 통해 공간 분석 및 재난 상태를 예측할 수 있어 향후 공간 스캐닝 분야와 관련된 여러 기술 분야에서 적극적으로 재사용될 수 있음.
- SOC 시설물 상태 및 현황과 관련하여 사회 안전을 위해 필요한 데이터를 실시간으로 공유할 수 있는 데이터 보급 체제는 국가안전 관리에도 도움이 되지만 기후, 사회 현상, 교통 등과 연계하여 새로운 응용 분야를 창출 할 수 있음. 정부 3.0 정책과 연계된 다양 틈새시장과 공공 안전 확보 가능성이 높음.
- 복구 위주의 재난 관리 체제를 예방 위주의 관리 체제로 변화시키며, 특히 민간 및 공공에서 위험 예측 모델을 저작할 수 있는 위험 정의 도구를 제공함으로써 다양한 시설물 관리 주체들이 재난이나 위험을 사전에 예측할 수 있음.
- 실시간 제어 및 계측을 위한 네트워크 구조를 다양한 분야에서 쉽고 유연하게 적용할 수 있도록 표준 인터페이스를 고안하여 다양한 시설물 관리 주체들이 활용할 수 있을 뿐만 아니라 공유를 통해 단일 관점에서 다양한 정보를 추출할 수 있게 함.
- 선형 시설물(제방)이나 대형 시설물(도로, 교량)의 경우 시설물의 GPS 형태의 위치 정보만으로는 정확한 관리를 할 수 없음. 제방의 경우 시점/종점의 거리가 길며, 직선이 아니므로 붕괴등 손상 발생 시 정확한 위치를 찾을 수 없음
- 따라서 시설물의 위치뿐 아니라 시설물 내의 위치관리는 정확한 유지관리 및 진단을 위해서 필수적인 요소임(공항, 철도역사 등 실내 네비게이션의 필요성)
- 기존 유비쿼터스 및 IoT 기반 원격 시설물 모니터링 시스템의 연구는 단순한 시설물의 포인트(위치) 관리에 대한 연구라고 할 수 있으며, 본 연구는 시설물내의 선형(Line), 면형(Area) 등 입체적 위치 관리가 가능한 기술을 개발하여 시설물내의 손상 발생시 그 영향이 인근에 미치는 영향 분석도 가능할 수 있도록 함.
- 이기종의 시설물과 관련한 다양한 정보, 재난재해 정보를 공유하고, 의사결정을 수립할 수 있도록 관련 메타데이터를 3차원 건설공간정보 기반으로 통합하여 운용되기 때문에 이를 근거로 정부 정책 자료 등에 반영(피드백)함으로써 관련 기술의 획기적인 발전 유도
- 주요 SOC 시설물뿐만 아니라 향후 공공청사, 대학 캠퍼스, 주상복합 단지 등의 다양한 대규모 수용 시설에도 직접적인 확장 적용이 가능하여, 기술적 확장성 및 파급효과가 클 것으로 예상

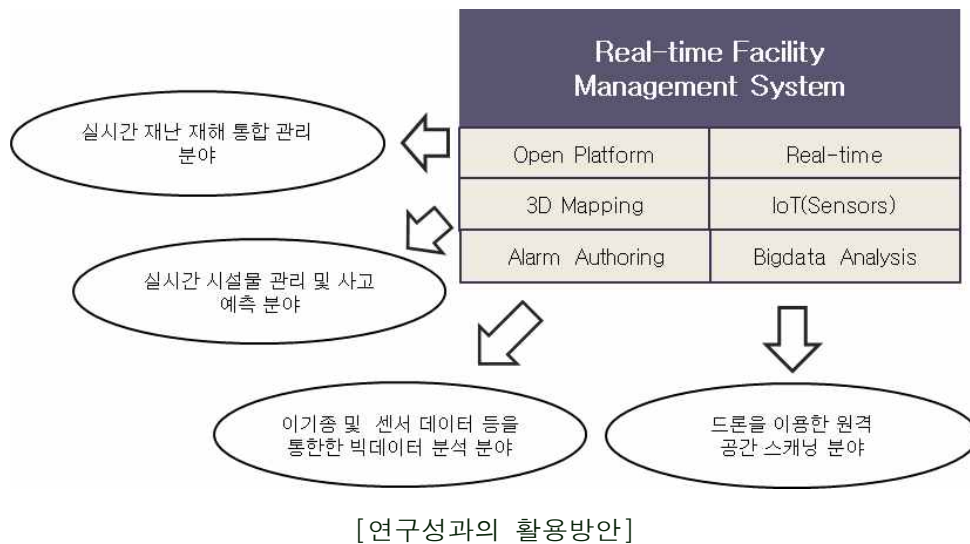
3. 목표 미달성 시 원인(사유) 및 차후대책(후속연구의 필요성 등)

○ 해당사항 없음

제5장. 연구개발성과의 활용계획

1. 활용 계획

- 실시간 감시가 필요한 인명이나 재산 피해가 예상되는 재난 예상 지역에 대한 현황 및 위험 예측 분야
(예) 사면, 도로, 지진, 산불 등
- 실시간 감시가 필요한 공공 시설물에 대한 이력, 현황, 위험 예측 분야
(예) 발전소, 변전소, 정유시설, 교량, 터널 등)
- 정부 3.0의 다양한 공공 정보 및 사회 정보를 통합하여 빅데이터 분석을 통한 의사결정 및 계획이 요구되는 분야
- 센서 및 IoT 기술 기반의 시설물 관리를 응용하기 위한 표준 인터페이스로 활용
- 드론을 이용한 원격 공간 스캐닝 응용 분야



- 2017년 11월 15일 포항 지진(규모 5.4) 등 최근 대규모의 지진 발생이 점차 증가하는 추세에 있으며, 지진 발생 시 광범위한 지역에 걸쳐서 막대한 인명피해와 경제적 손실을 일으키며, 국가 신경망의 마비 등 국가적 재난상황을 초래하고 있고, 특히 자연 재해의 경우 SOC 시설물 등의 2차 재난으로 연결되는 경우가 많아 예측되는 재난 유형에 따른 자동화된 점검 및 관리 체계가 절실히 요구되어 추가적인 연구가 필요함.

2. 현장 적용 방안

① 시스템 커스터마이징

- 시스템의 범용성을 확대하기 위하여 지속적으로 커스터마이징 도구를 개발하여 활용 한다. 기본적으로 API 레벨에 대한 패키징을 기반으로 하되 XML 등 범용 언어 및 GUI 기반 커스터마이징 도구를 개발함. 현재 단일 플랫폼에 대한 테스트가 완료된 상황이므로 향후 추가적인 플랫폼 독립적인 시스템으로 개선하여 협력 업체 및 시설물 관리 주체가 직접 적용할 수 있도록 추진함.

② 협력 업체 공동 활용

- 현재 시스템의 커스터마이징 수준을 확대하여 협력 업체들이 시설물 관리 응용 분야에서 라이선스 기반으로 적용할 수 있도록 추진함. 현재 공동 추진의사 있는 협력 업체들과 협의하고 있으며 향후 지속적 확대를 위해 커스터마이징 수준 향상과 소프트웨어 시스템 품질 보증 과정을 병행함.

③ SOC 시설물 관리 시범 운영

- SOC 시설물 관리 요구되는 일부 기관과 시범 운영 및 연구 향상 계획에 대해 추진할 계획임. 1차적으로 도로 시설물의 사면 분야에 대해 수도권 민자 구간 고속도로 관리 부분에서 활용하기 위해 협의하고 있음. 실제 해당 기관과는 연구 진행 중에 자문으로 참여하여 요구사항을 함께 분석하였으며 사면 등의 가속도, 화재, 유고 상황에 대한 내용을 포함할 예정임.

제6장. 연구 과정에서 수집한 해외 과학기술 정보

○ 해당사항 없음

제7장. 연구개발성과의 보안등급

○ 해당사항 없음

제8장. 국가과학기술종합정보시스템에 등록한 연구시설·장비 현황

구입 기관	연구시설/ 연구장비명	규격 (모델명)	수량	구입 연월일	구입 가격 (천원)	구입처 (전화번호)	비고 (설치 장소)	NTIS장비 등록 번호
----------	----------------	-------------	----	--------	---------------	---------------	---------------	-----------------

○ 해당사항 없음

제9장. 연구개발과제 수행에 따른 연구실 등의 안전 조치 이행 실적

○ 해당사항 없음

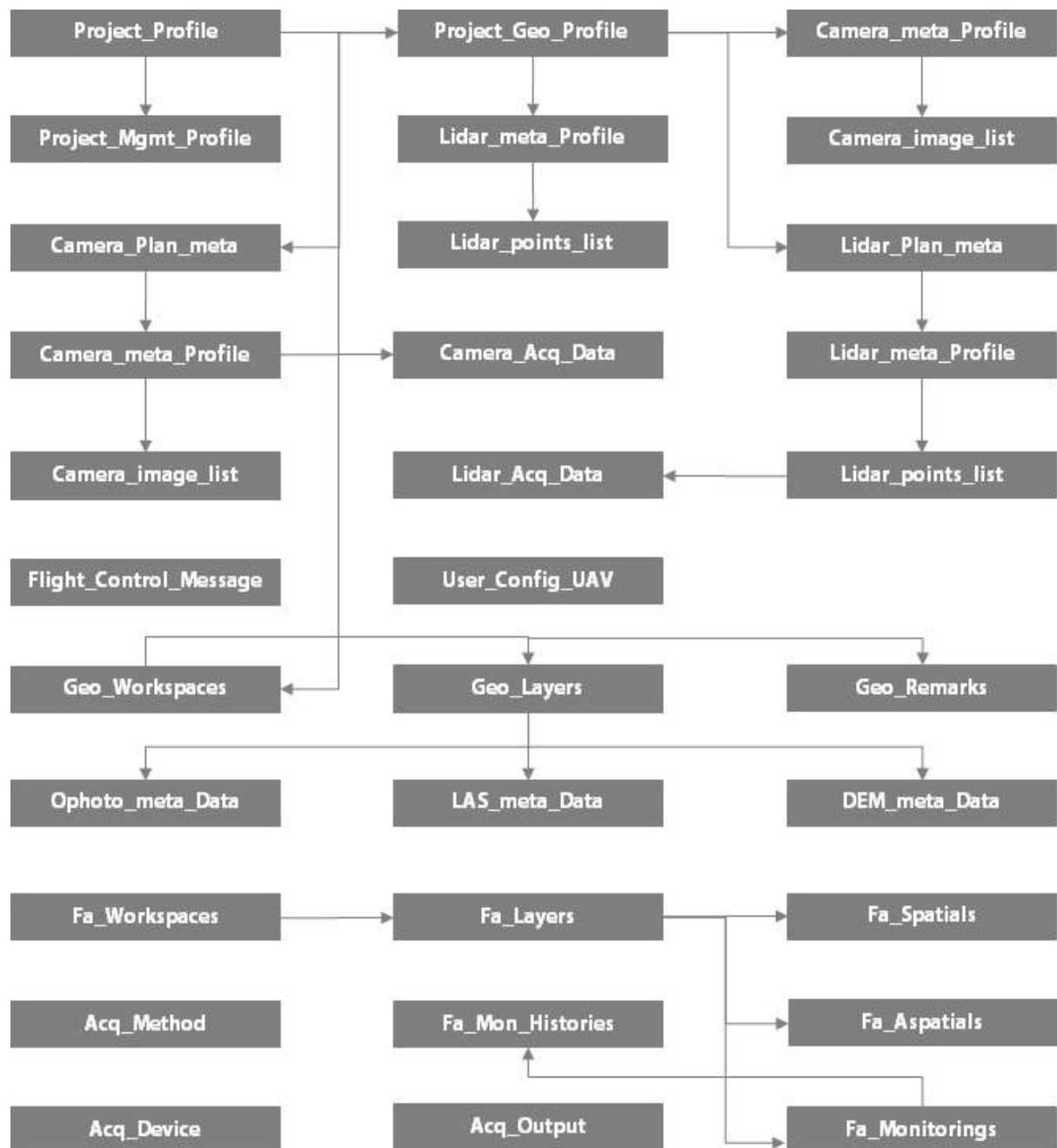
제10장. 연구개발과제의 대표적 연구 실적

번호	구분 (논문/ 특허/ 기타)	논문명/특허명/ 기타	소속 기관명	역할	논문 게재지/ 특허 등록 국가	영향력 지수	논문 게재일 /특허 등록일	사사 여부 (단독 사사 또는 중복 사사)	특기 사항 (SCI 여부/인용 횟수 등)
1	논문	A Process-Aware Drone-Equipped 3D Engine and Wireless Control Measurement Platform for Integrated Management of SOC Facilities	대림 대학교	주저자	ICACT Transactions on the Advanced Communications Technology		2017.11.30	단독사사	비SCI
2	논문	A process modeling tool for SOC facilities management process automation	대림 대학교	주저자	International Conference on Internet (ICONI)		2018.12.16	단독사사	비SCI
3	논문	A rule based file management tool for facility log files integration in disaster environments	대림 대학교	주저자	Journal of Internet Computing and Services(JICS)		2018.12.19	단독사사	비SCI
4	특허 출원	사회기반시설물 스마트 통합 유지 관리 방법	(주)케이지 아이		대한민국		2017.11.15	단독사사	
5	특허 출원	GPS위치기반 열화상 정보 분석 방법	(주)케이지 아이		대한민국		2018.09.17	단독사사	
6	프로그램 등록	Lidar(라이다) 정보를 이용한 Automatic(자동화) 3D 구현 엔진 시스템	(주)케이지 아이		대한민국		2017.11.27		
7	프로그램 등록	GPS 및 드론영상 기반 태양광 사전 경제성 분석 및 시설 관리	(주)케이지 아이		대한민국		2018.10.02		

제11장. 기타 사항

1. SOC 시설물 관리를 위한 DB 구조

과제에서 목표하고 있는 개발 시스템의 개념적 데이터베이스의 구조는 다음과 같다.



2. SOC 시설물 관리를 위한 개념적 DB 구조

(1) 공간 데이터 모델링 계획 편집

개념적 데이터베이스 설계	모듈명	GFMS/MART/PL		
	모듈개요	공간 데이터 모델링 계획 편집		
	작성일자	2018/09/23	버전	1.1
	책임자	이영걸	작성자	박민재



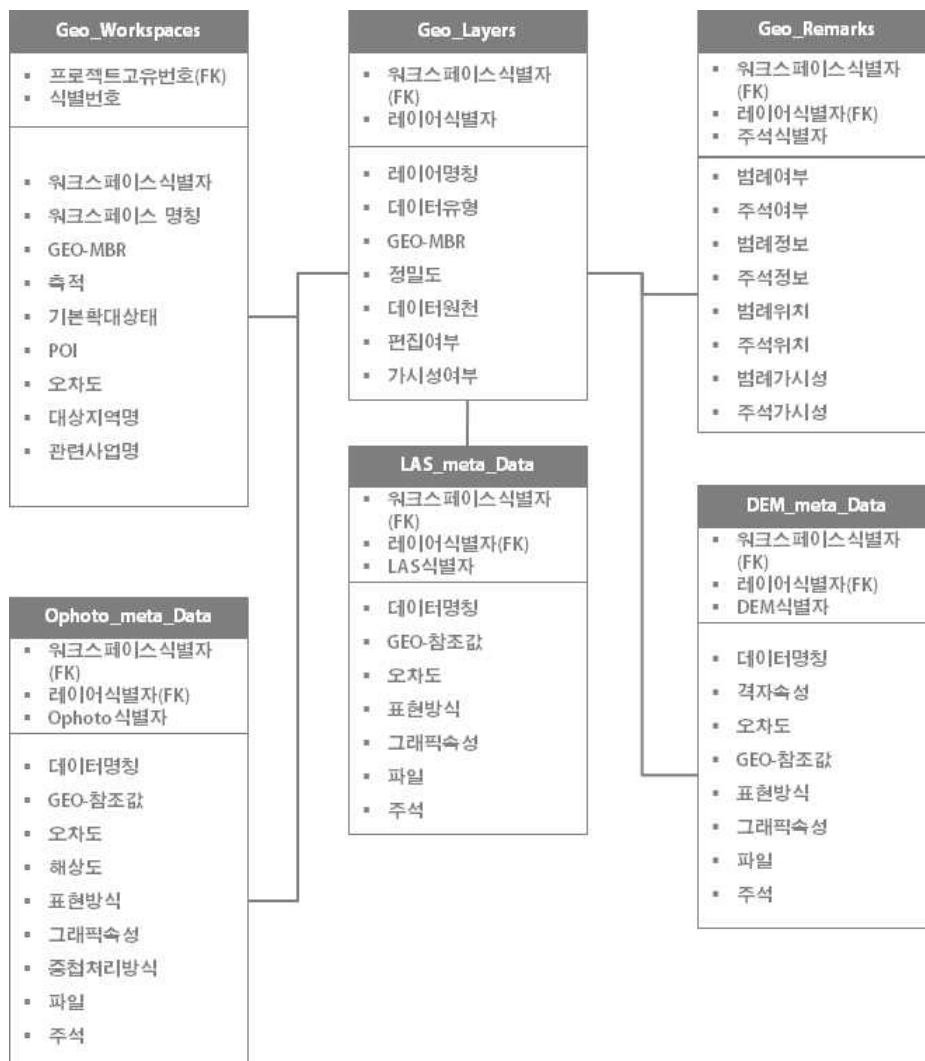
(2) 영상 및 라이다 데이터 획득

개념적 데이터베이스 설계	모듈명	GFMS/DART/DA		
	모듈개요	영상 및 라이다 데이터 획득		
	작성일자	2018/09/23	버전	1.1
	책임자	이영걸	작성자	박민재



(3) 공간 데이터 모델링 후처리

개념적 데이터베이스 설계	모듈명	GFMS/MART/GM		
	모듈개요	공간 데이터 모델링 후처리		
	작성일자	2018/09/23	버전	1.1
	책임자	이영걸	작성자	박민재



(4) 시설물 속성/모니터링 데이터 편집

개념적 데이터베이스 설계	모듈명	GFMS/MART/PL		
	모듈개요	시설물 속성/모니터링 데이터 편집		
	작성일자	2018/09/23	버전	1.1
	책임자	이영걸	작성자	박민재



(5) 서버 시스템 관리용 데이터베이스

개념적 데이터베이스 설계	모듈명	GFMS/MART/Admin		
	모듈개요	서버 시스템 관리용 데이터베이스		
	작성일자	2018/09/23	버전	1.1
	책임자	이영걸	작성자	박민재

Users
▪ 사용자식별자
▪ 비밀번호 ▪ 사용자명 ▪ 이메일 ▪ 사용자구분 ▪ 사용자그룹 ▪ 등록일자 ▪ 갱신일자 ▪ 활성화 ▪ 주석정보

Models
▪ 일련번호
▪ 대상구분 ▪ 대상명 ▪ 기본값 ▪ 생성자 ▪ 생성일시 ▪ 갱신일시

Diagnosis
▪ 일련번호
▪ 대상구분 ▪ 대상명 ▪ 진단조건 ▪ 기록방식 ▪ 생성자 ▪ 생성일시 ▪ 갱신일시

Groups
▪ 그룹식별자
▪ 그룹명 ▪ 그룹유형 ▪ 주석정보

Journals
▪ 일련번호
▪ 객체구분 ▪ 객체명 ▪ 사용자 ▪ 사용명령 ▪ 사용일시 ▪ 사용컴퓨터주소

Configurations
▪ 일련번호
▪ 설정대상구분 ▪ 설정변수명 ▪ 기본값 ▪ 생성자 ▪ 생성일시 ▪ 갱신일시

Privileges
▪ 권한식별자
▪ 객체구분 ▪ 객체명 ▪ 객체생성자 ▪ 부여자 ▪ 할당자 ▪ 부여권한 ▪ 부여권한사용여부

Constraints
▪ 일련번호
▪ 객체구분 ▪ 객체명 ▪ 생성자 ▪ 생성일시 ▪ 갱신일시 ▪ 제약조건구분 ▪ 제약조건

Agents
▪ 일련번호
▪ 에이전트명 ▪ 매크로 ▪ 타임스케줄 ▪ 생성자 ▪ 생성일시 ▪ 갱신일시 ▪ 주석정보

(6) 서버 시스템 환경용 데이터베이스

개념적 데이터베이스 설계	모듈명	GFMS/MART/SysInfo		
	모듈개요	서버 시스템 환경용 데이터베이스		
	작성일자	2018/09/23	버전	1.1
	책임자	이영걸	작성자	박민재

Projects
- 일련번호 - 프로젝트명 - 생성자명 - 경로명 - 프로젝트구분 - 생성일시 - 갱신일시 - 활성화 - 주석정보

Objects
- 일련번호 - 객체구분 - 객체명 - 객체통계 - 그룹유형 - 생성일시 - 갱신일시 - 활성화 - 주석정보

Domains
- 일련번호 - 도메인명 - 데이터타입매크로 - 생성일시 - 갱신일시 - 활성화 - 주석정보

Indexes
- 일련번호 - 대상객체 - 유니크값 - 색인방식 - 색인명 - 색인파일 - 생성자명 - 생성일시 - 갱신일시 - 활성화 - 주석정보

Monitorings
- 일련번호 - 대상객체구분 - 대상객체 - 모니터링명 - 모니터링매크로 - 생성자명 - 생성일시 - 갱신일시

Acquisitions
- 일련번호 - 획득방식구분 - 획득방식명 - 생성자 - 생성일시 - 갱신일시 - 주석정보

Devices
- 일련번호 - 장비구분 - 장비코드 - 장비명 - 가능획득방식 - 생성자명 - 생성일시 - 갱신일시

C_FOV
- 일련번호 - 카메라코드 - 카메라명 - FOV - 생성자명 - 생성일시 - 갱신일시

L_FOV
- 일련번호 - 라이다코드 - 라이다명 - FOV - 생성자명 - 생성일시 - 갱신일시

3. SOC 시설물 관리를 위한 논리적 DB 구조

(1) 프로젝트 프로파일(Project_Profile) 테이블

번호	속성명(국문)	속성명(영문)	데이터형식	제약조건	비고
1	프로젝트고유번호	project_id	문자열		기본키
2	프로젝트유형	project_type	문자열		
3	프로젝트이름	project_name	문자열		
4	프로젝트영문이름	project_ename	문자열		
5	프로젝트생성자	project_creator	문자열		
6	프로젝트생성일자	project_s_date	날짜시간		
7	프로젝트종료일자	project_e_date	날짜시간		
8	프로젝트경로	project_path	문자열		
9	주석정보	remarks	문자열		
10					
11					
12					
부연 설명					

(2) 프로젝트 GEO 프로파일(Project_Geo_Profile) 테이블

번호	속성명(국문)	속성명(영문)	데이터형식	제약조건	비고
1	프로젝트고유번호	project_id	문자열		기본키, 외래키
2	일련번호	serial_no	정수		기본키
3	작업지형영역	work_area	다각선		
4	MBR	work_area_mbr	사각형		
5	영역구분	area_type	문자열		
6	기본도중첩상태	map_over_st	정수		
7	영상중첩상태	img_over_st	정수		
8	라이다중첩상태	Ldr_over_st	정수		
9	열화상중첩상태	thr_over_st	정수		
10	주석정보	remarks	문자열		
11					
12					
부연 설명					

(3) 카메라_메타정보_프로파일(Camera_meta_Profile) 테이블

번호	속성명(국문)	속성명(영문)	데이터형식	제약조건	비고
1	프로젝트고유번호	project_id	문자열		기본키, 외래키
2	일련번호	serial_no	정수		기본키
3	카메라 유형	camera_type	문자열		
4	기본고도	b_height	정수		
5	기본FOV	b_fov	정수집합		
6	해상도	b_resolution	정수집합		
7	촬영일자	c_date	날짜시간		
8	촬영자	c_person	문자열		
9	주석정보	remarks	문자열		
10					
11					
12					
부연 설명					

(4) 프로젝트_관리_프로파일(Project_Mgmt_Profile) 테이블

번호	속성명(국문)	속성명(영문)	데이터형식	제약조건	비고
1	프로젝트고유번호	project_id	문자열		기본키, 외래키
2	일련번호	serial_no	정수		기본키
3	프로젝트관리유형	prj_mg_type	문자열		
4	영상데이터획득방식	img_cap_type	정수		
5	카메라기본고도	cam_height	정수		
6	카메라FOV	cam_fov	정수집합		
7	카메라중첩률	cam_overlay	정수		
8	라이다기본고도	ldr_height	정수		
9	라이다FOV	ldr_fov	정수집합		
10	라이다중첩률	ldr_overlay	정수		
11	열화상기본고도	thr_height	정수		
12	열화상FOV	thr_fov	정수집합		
13	열화상중첩률	thr_overlay	정수		
14	주석정보	remarks	문자열		
부연 설명					

(5) 라이다_메타정보_프로파일(Lidar_meta_Profile) 테이블

번호	속성명(국문)	속성명(영문)	데이터형식	제약조건	비고
1	프로젝트고유번호	project_id	문자열		기본키, 외래키
2	일련번호	serial_no	정수		기본키
3	라이다 유형	ldr_type	문자열		
4	기본고도	ldr_height	정수		
5	기본FOV	ldr_fov	정수집합		
6	주파수	ldr_frequency	정수		
7	촬영일자	ldr_s_date	날짜시간		
8	촬영자	ldr_s_person	문자열		
9	주석정보	remarks	문자열		
10					
11					
12					
13					
14					
부연 설명					

(6) 카메라_영상_목록(Camera_image_list) 테이블

번호	속성명(국문)	속성명(영문)	데이터형식	제약조건	비고
1	프로젝트고유번호	project_id	문자열		기본키, 외래키
2	일련번호	serial_no	정수		기본키
3	고도	height	정수		
4	위치	location	좌표		
5	FOV	fov	정수집합		
6	타일	tile	사각형		
7	촬영일자	s_date	날짜시간		
8	파일	cam_file	파일		
9	주석정보	remarks	문자열		
10					
11					
12					
13					
14					
부연 설명					

(7) 카메라_계획_메타정보(Camera_Plan_meta) 테이블

번호	속성명(국문)	속성명(영문)	데이터형식	제약조건	비고
1	프로젝트고유번호	project_id	문자열		기본키, 외래키
2	일련번호	serial_no	정수		기본키
3	계획번호	cam_plan_id	문자열		
4	계획명	cam_plan_name	문자열		
5	비행경로집합	route_path	좌표집합		
6	획득활동집합	acti_set	명령집합		
7	획득시작일시	cam_s_date	날짜시간		
8	획득종료일시	cam_e_date	날짜시간		
9	주석정보	remarks	문자열		
10					
11					
12					
13					
14					
부연 설명					

(8) 라이다_포인트_목록(Lidar_points_list) 테이블

번호	속성명(국문)	속성명(영문)	데이터형식	제약조건	비고
1	프로젝트고유번호	project_id	문자열		기본키, 외래키
2	일련번호	serial_no	정수		기본키
3	고도	height	정수		
4	위치	location	좌표값		
5	FOV	fov	정수집합		
6	버퍼	buffer	바이트집합		
7	촬영일자	s_date	날짜시간		
8	파일	ldr_file	파일		
9	주석정보	remarks	문자열		
10					
11					
12					
13					
14					
부연 설명					

(9) 라이다_계획_메타정보(Lidar_Plan_meta) 테이블

번호	속성명(국문)	속성명(영문)	데이터형식	제약조건	비고
1	프로젝트고유번호	project_id	문자열		기본키, 외래키
2	일련번호	serial_no	정수		기본키
3	계획번호	ldr_plan_id	문자열		
4	계획명	ldr_plan_name	문자열		
5	비행경로집합	route_path	좌표집합		
6	획득활동집합	acti_set	명령집합		
7	획득일자	ldr_s_date	날짜시간		
8	주석정보	remarks	문자열		
9					
10					
11					
12					
13					
14					
부연 설명					

(10) 카메라_메타정보_프로필(Camera_meta_Profile) 테이블

번호	속성명(국문)	속성명(영문)	데이터형식	제약조건	비고
1	프로젝트고유번호	project_id	문자열		기본키, 외래키
2	일련번호	serial_no	정수		기본키
3	카메라 유형	camera_type	문자열		
4	기본고도	b_height	정수		
5	기본FOV	b_fov	정수집합		
6	해상도	resolution	정수집합		
7	촬영일자	cam_c_date	날짜시간		
8	촬영자	remarks	문자열		
9	주석정보	remarks	문자열		
10					
11					
12					
13					
14					
부연 설명					

(11) 카메라_획득_데이터(Camera_Acq_Data) 테이블

번호	속성명(국문)	속성명(영문)	데이터형식	제약조건	비고
1	프로젝트고유번호	project_id	문자열		기본키, 외래키
2	일련번호	serial_no	정수		기본키
3	고도	cam_height	정수		
4	위치	cam_location	좌표값		
5	FOV	cam_fov	정수집합		
6	타일	cam_tile	사각형		
7	촬영일자	cam_c_date	날짜시간		
8	파일	cam_file	파일		
9	주석정보	remarks	문자열		
10					
11					
12					
13					
14					
부연 설명					

(12) 카메라_영상_목록(Camera_image_list) 테이블

번호	속성명(국문)	속성명(영문)	데이터형식	제약조건	비고
1	프로젝트고유번호	project_id	문자열		기본키, 외래키
2	일련번호	serial_no	정수		기본키
3	고도	c_heigjt	정수		
4	위치	c_location	좌표값		
5	FOV	c_gov	정수집합		
6	타일	c_tile	사각형		
7	촬영일자	c_c_date	날짜시간		
8	파일	c_file	파일		
9	주석정보	remarks	문자열		
10					
11					
12					
13					
14					
부연 설명					

(13) 라이다_획득_데이터(Lidar_Acq_Data) 테이블

번호	속성명(국문)	속성명(영문)	데이터형식	제약조건	비고
1	프로젝트고유번호	project_id	문자열		기본키, 외래키
2	일련번호	serial_no	정수		기본키
3	고도	ldr_height	정수		
4	위치	ldr_location	좌표값		
5	FOV	ldr_fov	정수집합		
6	버퍼	ldr_buffer	바이트집합		
7	촬영일자	ldr_s_date	날짜시간		
8	파일	ldr_file	파일		
9	주석정보	remarks	문자열		
10					
11					
12					
13					
14					
부연 설명					

(14) 라이다_포인트_목록(Lidar_points_list) 테이블

번호	속성명(국문)	속성명(영문)	데이터형식	제약조건	비고
1	프로젝트고유번호	project_id	문자열		기본키, 외래키
2	일련번호	serial_no	정수		기본키
3	고도	ld_heighr	정수		
4	위치	ld_location	좌표값		
5	FOV	lf_fov	정수집합		
6	버퍼	ld_buffer	바이트집합		
7	촬영일자	ld_s_date	날짜시간		
8	파일	ld_file	파일		
9	주석정보	remarks	문자열		
10					
11					
12					
13					
14					
부연 설명					

(15) 비행_제어_메시지(Flight_Control_Message) 테이블

번호	속성명(국문)	속성명(영문)	데이터형식	제약조건	비고
1	식별번호	project_id	문자열		기본키
2	명령코드	serial_no	바이트		
3	명령어	ld_heighr	문자열		
4	명령어설명	ld_location	문자열		
5	제약조건	lf_fov	문자열		
6	버전	ld_buffer	문자열		
7	주석정보	remarks	문자열		
8					
9					
10					
11					
12					
13					
14					
부연 설명					

(16) 비행_사용자_환경(User_Config_UAV) 테이블

번호	속성명(국문)	속성명(영문)	데이터형식	제약조건	비고
1	식별번호	project_id	문자열		기본키
2	통신속도	baud_rate	정수		
3	타임스탬프	timestamp	정수		
4	방향성	orientation	정수		
5	가속도	acceleration	정수		
6	속도	velocity	정수		
7	각속도	anglar_velo	정수		
8	위치	location	정수		
9	짐벌	gimbal	정수		
10	비행상태	flight_status	정수		
11	제어정보	control_info	문자열		
12					
13					
14					
부연 설명					

(17) 지오_작업공간(Geo_Workspaces) 테이블

번호	속성명(국문)	속성명(영문)	데이터형식	제약조건	비고
1	프로젝트고유번호	project_id	문자열		기본키, 외래키
2	일련번호	serial_no	정수		기본키
3	워크스페이스식별자	wk_id	정수		
4	워크스페이스 명칭	wk_name	문자열		
5	GEO-MBR	wk_geo_mbr	사각형		
6	축적	wk_scale	정수		
7	기본확대상태	wk_b_zoom	정수		
8	POI	wk_poi	좌표값		
9	오차도	wk_falv	정수		
10	대상지역명	wk_area_name	문자열		
11	관련사업명	wk_project_name	문자열		
12	주석정보	remarks	문자열		
13					
14					
부연 설명					

(18) 지오_레이어(Geo_Layers) 테이블

번호	속성명(국문)	속성명(영문)	데이터형식	제약조건	비고
1	프로젝트고유번호	project_id	문자열		기본키, 외래키
2	일련번호	serial_no	정수		기본키
3	레이어식별자	layer_id	정수		기본키
4	레이어명칭	layer_name	문자열		
5	데이터유형	data_src_type	문자열		
6	GEO-MBR	layer_mbr	사각형		
7	정밀도	layer_precision	정수		
8	데이터원천	data_source	문자열		
9	편집여부	editable	정수		
10	가시성여부	visible	정수		
11	주석정보	remarks	문자열		
12					
13					
14					
부연 설명					

(19) 지오_주석(Geo_Remarks) 테이블

번호	속성명(국문)	속성명(영문)	데이터형식	제약조건	비고
1	프로젝트고유번호	project_id	문자열		기본키, 외래키
2	일련번호	serial_no	정수		기본키
3	주석식별자	rem_id	문자열		기본키
4	범례여부	reg_exist	정수		
5	주석여부	rem_exist	정수		
6	범례정보	reg_info	바이트집합		
7	주석정보	rem_info	바이트집합		
8	범례위치	reg_loc	좌표값		
9	주석위치	rem_loc	좌표값		
10	범례가시성	reg_visible	정수		
11	주석가시성	rem_visible	정수		
12	주석정보	remarks	문자열		
13					
14					
부연 설명					

(20) 정사영상_메타정보_데이터(Ophoto_meta_Data) 테이블

번호	속성명(국문)	속성명(영문)	데이터형식	제약조건	비고
1	프로젝트고유번호	project_id	문자열		기본키, 외래키
2	일련번호	serial_no	정수		기본키
3	Ophoto식별자	ophoto_id	문자열		기본키
4	데이터명칭	data_name	문자열		
5	GEO-참조값	geo_ref	좌표값		
6	오차도	fal_val	정수		
7	해상도	resolution	정수집합		
8	표현방식	color_exp	문자열		
9	그래픽속성	graphic_att	바이트집합		
10	중첩처리방식	overlay_exp	정수		
11	파일	file	파일		
12	주석정보	remarks	문자열		
13					
14					
부연 설명					

(21) LAS_메타정보_데이터(LAS_meta_Data) 테이블

번호	속성명(국문)	속성명(영문)	데이터형식	제약조건	비고
1	프로젝트고유번호	project_id	문자열		기본키, 외래키
2	일련번호	serial_no	정수		기본키
3	LAS식별자	las_id	문자열		기본키
4	데이터명칭	data_name	문자열		
5	GEO-참조값	geo_ref	좌표값		
6	오차도	fal_val	정수		
7	표현방식	color_exp	문자열		
8	그래픽속성	graphic_att	바이트집합		
9	파일	overlay_exp	파일		
10	주석정보	remarks	문자열		
11					
12					
13					
14					
부연 설명					

(22) DEM_메타정보_데이터(LAS_meta_Data) 테이블

번호	속성명(국문)	속성명(영문)	데이터형식	제약조건	비고
1	프로젝트고유번호	project_id	문자열		기본키, 외래키
2	일련번호	serial_no	정수		기본키
3	DEM식별자	dem_id	문자열		기본키
4	데이터명칭	data_name	문자열		
5	격자속성	cell_att	바이트집합		
6	오차도	fal_val	정수		
7	GEO-참조값	geo_ref	좌표값		
8	표현방식	color_exp	문자열		
9	그래픽속성	graphic_att	바이트집합		
10	파일	remarks	파일		
11	주석정보	remarks	문자열		
12					
13					
14					
부연 설명					

(23) 시설물_작업영역(Fa_Workspaces) 테이블

번호	속성명(국문)	속성명(영문)	데이터형식	제약조건	비고
1	프로젝트고유번호	project_id	문자열		기본키, 외래키
2	식별번호	serial_no	정수		기본키
3	워크스페이스식별자	fwk_id	문자열		기본키
4	워크스페이스 명칭	fwk_name	문자열		
5	GEO-MBR	geo_ref	좌표값		
6	레이어정보	layer_info	바이트집합		
7	시설물관리명칭	fa_mg_name	문자열		
8	관리책임자	fg_mg_charger	문자열		
9	주석정보	remarks	문자열		
10					
11					
12					
13					
14					
부연 설명					

(24) 시설물_레이어(Fa_Layers) 테이블

번호	속성명(국문)	속성명(영문)	데이터형식	제약조건	비고
1	워크스페이스식별자(FK)	fwk_id	문자열		기본키, 외래키
2	레이어식별자	flay_id	문자열		기본키
3	레이어명칭	flay_name	문자열		
4	데이터유형	data_type	문자열		
5	GEO-MBR	geo_ref	좌표값		
6	데이터원천	data_source	문자열		
7	편집여부	editable	정수		
8	가시성여부	visible	정수		
9	관리자정보	mgm_info	문자열		
10	주석정보	remarks	문자열		
11					
12					
13					
14					
부연 설명					

(25) 시설물_공간데이터(Fa_Spatials) 테이블

번호	속성명(국문)	속성명(영문)	데이터형식	제약조건	비고
1	워크스페이스식별자(FK)	fwk_id	문자열		기본키, 외래키
2	레이어식별자	flay_id	문자열		기본키, 외래키
3	객체식별자	obj_id	문자열		기본키
4	공간정보	spa_data	바이트집합		
5	그래픽속성	spa_graphic	바이트집합		
6	주석정보	remarks	문자열		
7					
8					
9					
10					
11					
12					
13					
14					
부연 설명					

(26) 시설물_비공간데이터(Fa_Aspatials) 테이블

번호	속성명(국문)	속성명(영문)	데이터형식	제약조건	비고
1	워크스페이스식별자(FK)	fwk_id	문자열		기본키, 외래키
2	레이어식별자	flay_id	문자열		기본키, 외래키
3	객체식별자	obj_id	문자열		기본키
4	기본속성	b_attribute	바이트집합		
5	사용자정의속성	u_attribute	바이트집합		
6	주석정보	remarks	문자열		
7					
8					
9					
10					
11					
12					
13					
14					
부연 설명					

(27) 시설물_모니터링데이터(Fa_Monitorings) 테이블

번호	속성명(국문)	속성명(영문)	데이터형식	제약조건	비고
1	워크스페이스식별자(FK)	fwk_id	문자열		기본키, 외래키
2	레이어식별자	flay_id	문자열		기본키, 외래키
3	객체식별자	obj_id	문자열		기본키
4	일련번호	seq_id	정수		
5	모니터링구분	mon_type	정수		
6	모니터링방식	mon_mode	정수		
7	모니터링주기	mon_prd	바이트집합		
8	모니터링속성	mon_attribute	문자열		
9	장치구분	dev_type	문자열		
10	출력방식	ouput_mode	문자열		
11	결과용공간데이터	result_data	바이트집합		
12	주석정보	remarks	문자열		
13					
14					
부연 설명					

(28) 시설물_모니터링_이력데이터(Fa_Mon_Histories) 테이블

번호	속성명(국문)	속성명(영문)	데이터형식	제약조건	비고
1	워크스페이스식별자(FK)	fwk_id	문자열		기본키, 외래키
2	레이어식별자	flay_id	문자열		기본키, 외래키
3	객체식별자	obj_id	문자열		기본키, 외래키
4	모니터링식별자	mon_id	문자열		기본키
5	일련번호	seq	정수		기본키
6	모니터링구분	mon_type	정수		
7	모니터링일시	mon_date	날짜구분		
8	출력데이터	output_data	바이트집합		
9	관리자	manager	문자열		
10	주석정보	remarks	문자열		
11					
12					
13					
14					
부연 설명					

(29) 시설물_데이터획득_방식(Acq_Method) 테이블

번호	속성명(국문)	속성명(영문)	데이터형식	제약조건	비고
1	식별자	id	문자열		기본키
2	데이터획득코드	da_code	문자열		
3	획득방식	da_method	문자열		
4	제약조건	da_constraint	문자열		
5	주석정보	remarks	문자열		
6					
7					
8					
9					
10					
11					
12					
13					
14					
부연 설명					

(30) 시설물_데이터획득_장치(Acq_Device) 테이블

번호	속성명(국문)	속성명(영문)	데이터형식	제약조건	비고
1	식별자	id	문자열		기본키
2	데이터획득장치코드	dad_code	문자열		
3	장치명	dad_name	문자열		
4	제약조건	dad_constraint	문자열		
5	주석정보	remarks	문자열		
6					
7					
8					
9					
10					
11					
12					
13					
14					
부연 설명					

(31) 시설물_데이터획득_출력방식(Acq_Output) 테이블

번호	속성명(국문)	속성명(영문)	데이터형식	제약조건	비고
1	식별자	id	문자열		기본키
2	출력방식코드	dar_code	문자열		
3	출력방식명	dar_name	문자열		
4	제약조건	dar_constraint	문자열		
5	주석정보	remarks	문자열		
6					
7					
8					
9					
10					
11					
12					
13					
14					
부연 설명					

(32) 사용자_정보(Users) 테이블

번호	속성명(국문)	속성명(영문)	데이터형식	제약조건	비고
1	사용자식별자	id	문자열		기본키
2	비밀번호	passwd	문자열		
3	사용자명	user_name	문자열		
4	이메일	email	문자열		
5	사용자구분	user_category	문자열		
6	사용자그룹	user_group	문자열		
7	등록일자	create_date	날짜시간		
8	갱신일자	update_date	날짜시간		
9	활성화	activate	정수		
10	주석정보	remarks	문자열		
11					
12					
13					
14					
부연 설명					

(33) 그룹_정보(Groups) 테이블

번호	속성명(국문)	속성명(영문)	데이터형식	제약조건	비고
1	그룹식별자	group_id	문자열		기본키
2	그룹명	group_name	문자열		
3	그룹유형	group_category	문자열		
4	주석정보	remarks	문자열		
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
부연 설명					

(34) 권한_정보(Privileges) 테이블

번호	속성명(국문)	속성명(영문)	데이터형식	제약조건	비고
1	권한식별자	prv_id	문자열		기본키
2	객체구분	prv_type	문자열		
3	객체명	object_name	문자열		
4	객체생성자	object_creator	문자열		
5	부여자	grantor	문자열		
6	할당자	grantee	문자열		
7	부여권한	peromission	문자열		
8	부여권한사용여부	usable	문자열		
9	주석정보	remarks	문자열		
10					
11					
12					
13					
14					
부연 설명					

(35) 로깅_정보(Journals) 테이블

번호	속성명(국문)	속성명(영문)	데이터형식	제약조건	비고
1	일련번호	prv_id	문자열		기본키
2	객체구분	prv_type	문자열		
3	객체명	object_name	문자열		
4	사용자	object_creator	문자열		
5	사용명령	action	문자열		
6	사용일시	action_date	날짜시간		
7	사용컴퓨터주소	action_ip	문자열		
8	주석정보	remarks	문자열		
9					
10					
11					
12					
13					
14					
부연 설명					

(36) 제약조건_정보(Constraints) 테이블

번호	속성명(국문)	속성명(영문)	데이터형식	제약조건	비고
1	일련번호	id	문자열		기본키
2	객체구분	object_type	문자열		
3	객체명	object_name	문자열		
4	생성자	creator	문자열		
5	생성일시	create_date	날짜시간		
6	제약조건구분	const_type	정수		
7	제약조건	const_macro	바이트집합		
8	주석정보	remarks	문자열		
9					
10					
11					
12					
13					
14					
부연 설명					

(37) 진단_정보(Diagnosis) 테이블

번호	속성명(국문)	속성명(영문)	데이터형식	제약조건	비고
1	일련번호	id	문자열		기본키
2	대상구분	target_type	문자열		
3	대상명	target_name	문자열		
4	진단조건	diag_condition	문자열		
5	기록방식	log_method	문자열		
6	생성자	creator	문자열		
7	생성일시	create_date	날짜시간		
8	갱신일시	update_date	날짜시간		
9	주석정보	remarks	문자열		
10					
11					
12					
13					
14					
부연 설명					

(38) 설정_정보(Configurations) 테이블

번호	속성명(국문)	속성명(영문)	데이터형식	제약조건	비고
1	일련번호	id	문자열		기본키
2	설정대상구분	target_type	문자열		
3	설정변수명	config_var_name	문자열		
4	기본값	def_val	바이트		
5	생성자	creator	문자열		
6	생성일시	create_date	날짜시간		
7	갱신일시	update_date	날짜시간		
8	주석정보	remarks	문자열		
9					
10					
11					
12					
13					
14					
부연 설명					

(39) 에이전트_정보(Agents) 테이블

번호	속성명(국문)	속성명(영문)	데이터형식	제약조건	비고
1	일련번호	id	문자열		기본키
2	에이전트명	agent_name	문자열		
3	매크로	macro	문자열		
4	타임스케줄	time_schedule	바이트집합		
5	생성자	creator	문자열		
6	생성일시	cretae_date	날짜시간		
7	갱신일시	update_date	날짜시간		
8	주석정보	remarks	문자열		
9					
10					
11					
12					
13					
14					
부연 설명					

(40) 프로젝트_정보(Projects) 테이블

번호	속성명(국문)	속성명(영문)	데이터형식	제약조건	비고
1	일련번호	prj_id	문자열		기본키
2	프로젝트명	prj_name	문자열		
3	생성자명	creator	문자열		
4	경로명	prj_path	문자열		
5	프로젝트구분	prj_type	문자열		
6	생성일시	create_date	날짜시간		
7	갱신일시	update_date	날짜시간		
8	활성화	activate	정수		
9	주석정보	remarks	문자열		
10					
11					
12					
13					
14					
부연 설명					

(40) 객체_정보(Objects) 테이블

번호	속성명(국문)	속성명(영문)	데이터형식	제약조건	비고
1	일련번호	obj_id	문자열		기본키
2	객체구분	obj_type	문자열		
3	객체명	obj_name	문자열		
4	객체통계	obj_stat	바이트집합		
5	그룹유형	group_type	문자열		
6	생성일시	create_date	날짜시간		
7	갱신일시	update_date	날짜시간		
8	활성화	activate	정수		
9	주석정보	remarks	문자열		
10					
11					
12					
13					
14					
부연 설명					

(41) 도메인_정보(Domains) 테이블

번호	속성명(국문)	속성명(영문)	데이터형식	제약조건	비고
1	일련번호	obj_id	문자열		기본키
2	도메인명	dom_name	문자열		
3	데이터타입매크로	macro	바이트집합		
4	생성일시	cretae_date	날짜시간		
5	갱신일시	update_date	날짜시간		
6	활성화	activate	정수		
7	주석정보	remarks	문자열		
8					
9					
10					
11					
12					
13					
14					
부연 설명					

(42) 모니터링_정보(Monitorings) 테이블

번호	속성명(국문)	속성명(영문)	데이터형식	제약조건	비고
1	일련번호	obj_id	문자열		기본키
2	대상객체구분	obj_type	문자열		
3	대상객체	obj_name	문자열		
4	모니터링명	mon_name	문자열		
5	모니터링매크로	mon_macro	문자열		
6	생성자명	creator	문자열		
7	생성일시	cretae_date	날짜시간		
8	갱신일시	update_date	날짜시간		
9	주석정보	remarks	문자열		
10					
11					
12					
13					
14					
부연 설명					

(43) 데이터획득_정보(Acquisitions) 테이블

번호	속성명(국문)	속성명(영문)	데이터형식	제약조건	비고
1	일련번호	obj_id	문자열		기본키
2	획득방식구분	da_method	문자열		
3	획득방식명	da_method_name	문자열		
4	생성자명	creator	문자열		
5	생성일시	cretae_date	날짜시간		
6	갱신일시	update_date	날짜시간		
7	주석정보	remarks	문자열		
8					
9					
10					
11					
12					
13					
14					
부연 설명					

(44) 데이터획득_장치정보(Devices) 테이블

번호	속성명(국문)	속성명(영문)	데이터형식	제약조건	비고
1	일련번호	dev_id	문자열		기본키
2	장비구분	dev_type	문자열		
3	장비코드	dev_code	문자열		
4	장비명	dev_name	문자열		
5	가능획득방식	dev_function	문자열		
6	생성자명	creator	문자열		
7	생성일시	create_date	날짜시간		
8	갱신일시	update_date	날짜시간		
9	주석정보	remarks	문자열		
10					
11					
12					
13					
14					
부연 설명					

(45) 카메라_FOV정보(C_FOV) 테이블

번호	속성명(국문)	속성명(영문)	데이터형식	제약조건	비고
1	일련번호	cf_id	문자열		기본키
2	카메라코드	cf_cam_code	문자열		
3	카메라명	cf_cam_name	문자열		
4	FOV	cf_fov	정수집합		
5	생성자명	creator	문자열		
6	생성일시	cretae_date	날짜시간		
7	갱신일시	update_date	날짜시간		
8	주석정보	remarks	문자열		
9					
10					
11					
12					
13					
14					
부연 설명					

(46) 라이다_FOV정보(L_FOV) 테이블

번호	속성명(국문)	속성명(영문)	데이터형식	제약조건	비고
1	일련번호	lf_id	문자열		기본키
2	라이다코드	lf_cam_code	문자열		
3	라이다명	lf_cam_name	문자열		
4	FOV	lf_fov	정수집합		
5	생성자명	creator	문자열		
6	생성일시	cretae_date	날짜시간		
7	갱신일시	update_date	날짜시간		
8	주석정보	remarks	문자열		
9					
10					
11					
12					
13					
14					
부연 설명					

제12장. 참고 문헌

1. 소방청, 초고층 및 지하연계 복합건축물 재난관리에 관한 특별법, 2017.
 2. Open GIS Consortium, Feature Spatial Schema(ISO 19107), 2001.
 3. Open GIS Consortium, Simple Feature Access(ISO 19125), 2011.
 4. buildingSMART, <http://www.buildingsmart-tech.org/ifc/IFC2x4/alpha/html/index.htm>.
 5. AutoCAD DXF 파일 구조,
<http://docs.autodesk.com/ACD/2011/KOR/filesDXF/WS1a9193826455f5ff18cb41610ec0a2e719-796b.htm>
 6. Tecplot Data Format Guide, http://download.tecplot.com/360/current/360_data_format_guide.pdf.
 7. 강태욱 외 3인, “BIM과 GIS간 정보상호운용을 위한 외부 BIM 참조 모델 제안”, 한국공간정보학회지 20권 5호, 2012
 8. 소방방재연구소, 초고층 건축물에 대한 방화 안전대책, 2015
 9. SH공사 도시연구소, SH공사 공동주택 건축구조설계지침, 2014
 10. 이동훈 외 5인, “전산플랫폼을 이용한 비정형 초고층 건축물 성능기반 내진설계기술의 실무적용”, 한국전산구조공학회 논문집 제26권 제1호, 2013
 11. 대한건축학회, BIM 설계도서 작성 지침(건축, 구조 부문), 2016
 12. 명지대학교 산학협력단, “BIM 실무 구현을 위한 구조화된 지식기반 통합객체분류 자동화”, 2014
 13. 김태권 외 1인, “화재시뮬레이션을 활용한 노래방 화재 위험성 분석”, 한국산학기술학회논문지 제15권 제5호, 2014
-

주 의

1. 이 최종보고서는 국토교통부에서 시행한 국토교통기술촉진연구사업의 연구보고서입니다.
2. 이 최종보고서 내용을 발표하는 때에는 반드시 국토교통부에서 시행한 사업의 연구개발성과임을 밝혀야 합니다.
3. 국가과학기술 기밀 유지에 필요한 내용은 대외적으로 발표 또는 공개하여서는 안 됩니다.