

최종보고서(일반과제용)							보안등급				
							일반[✓], 보안[]				
중앙행정기관명	국토교통부		사업명	사업명		국토교통기술촉진연구					
전문기관명	국토교통과학기술진흥원		내역사업명								
공고번호	공고-제2호 2019년 국토교통기술촉진연구사업		총괄연구개발 식별번호		21CTAP-C152294						
			연구개발과제번호		21CTAP-C152294-03						
국가과학기술 표준분류	1순위 EE0303	50%	2순위 EA0907	25%	3순위 EI0505	20%					
총괄연구개발명	국문	사회이슈해결 지원유형									
	영문	Social Issue Resolution Support Type									
연구개발과제명	국문	화물차 운전자(차량)의 안전운행 행위에 대한 보상을 위한, 블록체인 기반 디지털운행기록단말 장치와 안전운행 보상 시스템 개발									
	영문	Developed a blockchain based Digital Tachograph and a compensation system for safe driving to compensate for the safe driving behavior of a long distance truck driver(Vehicle)									
주관연구기관	기관명	주식회사 퀀텀게이트		사업자등록번호	748-87-00331						
	주소	세종특별자치시 연서면 도신고북로 1203 (우)30043		법인등록번호	141211-0081590						
주관연구책임자	성명	백주용		직위	대표이사						
	연락처	직장전화	033-766-2762		휴대전화	010-6209-2762					
		전자우편	v@quantumgate.co.kr		국가연구자번호	1106 - 9770					
연구개발기간	전체	2019. 04. 19 - 2021. 12. 31(2 년 8.5개월)									
	1년차	2019. 04. 19 - 2019. 12. 31(1 년 8.5개월)									
	2년차	2020. 01. 01 - 2020. 12. 31(1 년 12개월)									
	3년차	2021. 01. 01 - 2021. 12. 31(1 년 12개월)									
	m년차	-									
	n년차	-									
연구개발비 (단위 : 천원)	정부지원 연구개발비	기관부담 연구개발비		그외기관 등의지원금 자립자단체 기타()		합계			연구개발비 외 지원금		
	현금	현금	현물	현금	현물	현금	현물	합계			
총계	600,000	20,000	180,003	-	-	-	-	620,000	180,003	800,003	-
1년차	134,000	4,467	40,203	-	-	-	-	138,467	40,203	178,670	-
2년차	140,000	4,666	42,000	-	-	-	-	144,666	42,000	186,666	-
3년차	326,000	10,867	97,800	-	-	-	-	336,867	97,800	434,667	-
m년차	-	-	-	-	-	-	-	-	-	-	-
n년차	-	-	-	-	-	-	-	-	-	-	-
공동연구기관 등	기관명	책임자	직위	휴대전화	전자우편		비고				
							역할	기관유형			
공동연구기관	-	-	-	-	-		-	-			
위탁연구기관	상명대학교 산학협력단	김종원	교수	010-5433-9773	jwkim@smu.ac.kr		-	대학			
연구기관 외 기관	-	-	-	-	-		-	-			
연구개발담당자 실무담당자	성명	김도원		직위	전임연구원						
	연락처	직장전화	033-766-2762		휴대전화	010-2263-3873					
		전자우편	darkhose954@quantumgate.co.kr		국가연구자번호	1240 - 9143					

이 최종보고서에 기재된 내용이 사실임을 확인하며, 만약 사실이 아닌 경우 관련 법령 및 규정에 따라 제재 처분 등의 불이익도 감수하겠습니다.

2022년 02월 03일

주관연구책임자 : 백 주 용

주관연구기관의 장 : 주식회사 퀀텀게이트 백 주 용

국토교통과학기술진흥원장 귀하

본 서식은 「국가연구개발혁신법」 시행('21.01.01) 이전에 최종 협약하여 '21.01.01 이후에 최종평가를 시행하는 일반과제용 입니다.

< 요약 문 >

사업명		국토교통기술촉진연구				총괄연구개발 식별번호		21CTAP-C152294						
내역사업명						연구개발과제번호		21CTAP-C152294-03						
국가과학기술 표준분류		1순위 EE0303		50%	2순위 EA0907		25%	3순위 EI0505		20%				
총괄연구개발명		사회이슈해결 지원유형												
연구개발과제명		화물차 운전자(차량)의 안전운행 행위에 대한 보상을 위한, 블록체인 기반 디지털운행 기록단말 장치와 안전운행 보상 시스템 개발												
전체 연구개발기간		2019. 04. 19 - 2021. 12. 31(2년 9개월)												
총 연구개발비		총 800,010천원 (정부지원연구개발비 : 600,000천원, 기관부담연구개발비 : 200,010천원, 지방자치단체 : 0천원, 그 외 지원금 : 0천원)												
연구개발단계		기초[] 응용[] 개발[☒] 기타(위 3가지에 해당되지 않는 경우)[]				기술성숙도		착수시점 기준(2) 종료시점 목표(7)						
연구개발과제 유형		지정공모 [☒] 자유공모 []												
연구개발과제 특성		실용화												
연구개발 목표 및 내용	최종 목표	• 디지털운행기록장치(기본 Spec.)+ADAS(전방추돌, 차선이탈)+블록체인 기반 안전운행 보상 시스템이 하나로 결합된 All in One 제품개발 • DTG 단말용 블록체인 네트워크 및 블록체인기반의 안전운전 보상시스템 개발												
	전체 내용	• FHD 카메라 탑재 ADAS(전방추돌, 차선이탈) 개발로 안전운행을 지원하고 영상기능을 활용하여 안전운행 이행상황 검증용으로 개발(차선, 안전거리) • SDRS(Safe Driving Reward System)시스템의 안전운전 블록체인 네트워크 개발 • SDCN(Safe Driving Channel Network)안전채널 • SDRCN(Safe Driving Reward Channel Network) • DTG 단말기 데이터 위·변조 방지를 위한 라이트 블록체인 개발 • 유사환경 및 제한된 실제 환경에서 시스템의 측정 및 신뢰도 검증												
		1년차	목표	차량운행기록장치제작, 차량운행기록장치 시스템S/W개발, 블록체인보상시스템										
	2년차	내용	J1939/J1708 Dongle개발, 차량운행기록메인장치 개발,											
		목표	ADAS 장치제작, 영상처리 알고리즘개발, 블록체인 보상시스템											
	3년차	내용	영상처리장치H/W개발,전방차량,차선이탈,급차로변경,안전거리미확보,블록체인네트워크											
		목표	시제품 제작, 차량관제서버 개발, 시스템장치와 서버연동, 블록체인 보상시스템											
	3년차	내용	차량운행데이터 DB생성 및 연동,차량운행기록장치 연동, 애플리케이션 및 모바일 연동											
연구개발성과		1. 디지털운행기록단말기의 기술적인 측면의 기대성과 가. 차량운행기록장치(기본 Spec.)+ADAS(전방추돌, 차선이탈)+블록체인 기반 안전운행 보상 시스템 이 하나로 결합된 상품(신개념, 신제품) 개발 나. 운행기록 데이터의 위변조 불가능, 실시간 데이터 저장 및 분석 가능 다. 고화질카메라(FHD) 추가 장착으로 안전운행 보조기능 강화 라. 화물운전자나 대형업체에서 운행데이터를 업로드 할 필요 없이 LTE, 5G 통신으로 실시간 업로드 마. FHD카메라 추가 장착으로 안전운행 보조기능 강화 추가 개발 가능 1) 불법 유턴, 중앙선 침범, 지정 차로 위반, 차로 급 변경, 차선이탈, 안전거리 미확보 등의 운행 영상 분석을 통해 확인 및 경고 2) 상기 데이터를 수치화하여 블록체인 기반의 Dapp으로 저장 및 보상 3) 당사 보유기술인, RGB 카메라 기반 차량 및 차선검출 기술 활용 추가개발 4) RGB 카메라로 촬영한 도로 영상에서 검출속도 향상을 위한 차선 별 분할방법 연구												
	연구개발성과 활용방안 및 기대효과	1. 상용차 텔레매틱스 분야 진출 장거리 화물차 및 버스 우선진출 2. 한국교통연구원의 ‘View T 2.0’ 플랫폼에 적용 공동수익모델 창출 3. 국토부 ‘스마트시티통합플랫폼’과 연계하여 국내 및 해외진출 국가, 지자체의 장거리 화물차나 버스의 안전운행 보상지원 시스템 가능 4. 향후 관련법규, 기술적인문제, 보상체계 등이 해결되면 하이패스 단말기대체 가능												
		해당없음												
		연구개발성과의 등록·기탁 건수	논문	특허	보고서 원문	연구 시설·장비	기술 요약 정보	소프트웨어	표준	생명자원		화학물	신품종	
										생명 정보	생물 자원		정보	실물
연구시설·장비 종합정보시스템 등록 현황		구입기관	연구시설·장비명		규격 (모델명)	수량	구입 연월일	구입가격 (천원)	구입처 (전화)	비고 (설치장소)	ZEUS 등록번호			
국문핵심어 (5개 이내)		화물차		안전운행		디지털운행 기록장치		보상시스템		블록체인				
영문핵심어 (5개 이내)		Long Distance Truck		Safe Transportation		Digital Tachograph		Reward System		Blockchain				

〈 목 차 〉

1. 연구개발과제의 개요
2. 연구개발과제의 수행 과정 및 수행내용
3. 연구개발과제의 수행 결과 및 목표 달성 정도
4. 목표 미달 시 원인분석(해당 시 작성)
5. 연구개발성과 및 관련 분야에 대한 기여 정도
6. 연구개발성과의 관리 및 활용 계획

별첨자료(참고문헌 등)

2) 연구개발과제의 년차별 목표

○ 1차 년도

(1) 차량 J1939/J1708 시험환경구축

- 차량운행기록장치개발시 J1939/J1708 Dongle을 사용하여 차량으로부터 차량운행정보(속도, RPM, 브레이크신호)를 추출해야함
- 차량운행정보(속도, RPM, 브레이크신호)를 추출시 실차량에서 필요정보를 추출하는 것은 시간이 매우 많이 소요되고 번거로운 작업임
- 개발효율성 향상을 위해 실차량을 사용하지 않고 차량시뮬레이션장치를 사용하여 필요한 차량운행정보를 효율적으로 추출함

(2) J1939/J1708 Dongle 제작

- 차량운행기록장치에서 사용한 차량운행데이터(속도, RPM, 브레이크신호)를 추출하기 위한 장치 제작
- J1939/J1708 프로토콜에 맞추어 차량운행데이터를 추출함
- 추출된 차량운행데이터를 CAN통신을 사용하여 차량운행기록장치에 전송함

(3) 차량운행기록장치 제작

- CAN통신장치를 사용하여 J1939/J1708 Dongle로부터 전송된 차량운행데이터를 수신함
- 차량운행데이터에서 차량속도/분당 엔진 회전수(RPM)/브레이크신호를 검출 및 저장함
- GPS로부터 수신된 위치 정보를 수신 및 저장함
- 가속도센서를 이용하여 충격을 감지함
- 기억장치를 사용하여 6개월이상 1초 단위 데이터를 기록 및 저장함
- 사용자가 현 운행상태를 시각적으로 확인할수 있는 표시기를 사용함
- 사용자가 현 운행상태를 청각적으로 확인할수 있도록 음성출력기능 개발
- 입력장치를 사용하여 사용자 정보를 입력함

(4) 시험환경 데이터 수집 : 10일치 데이터 수집

(5) Virtual Machine 기반의 Private Blockchain Network Test bed 구축

- 프라이빗 블록체인 네트워크는 분산원장 네트워크로서 여러 대의 서버로 구성된 네트워크 솔루션 임.
- 따라서 각각의 서버의 환경에 적합한 OS와 개발툴, DBMS 등이 설치됨으로 소프트웨어 개발을 위한 환경구축에 많은 노력과 시간이 소요되는 것을 최적화해야 함.
- 이러한 네트워크 솔루션 개발과 배포, 유지에 적합한 가상머신(Virtual Machine)을 이용한 개발환경을 구축함으로써 효율성을 높이고자 함.

	Peer1	Peer2	Peer3	Peer4
				
OS	Linux	Linux	Linux	Linux
VM tool	docker	docker	docker	docker
Dev language	Go, Node. Java	Go, Node. Java	Go, Node. Java	Go, Node. Java
DBMS	couchDB	couchDB	couchDB	couchDB

그림 3 개발환경 가상화

(6) SDCS(Safety Driving Compensation System) DBMS 시스템

- SDCS DBMS 시스템은 기존에 사용되고 있는 DTG단말기의 운행정보를 SDCS서버에 업로드 함으로써 위험운전행동을 분석하고 도로 위치별 운전자별, 운수회사별 정보를 검색할 수 있는 시스템
- DTG 단말기에 저장되는 정보는 차량의 운행정보를 담고 있는 1초주기 저장 데이터와 차량의 충돌정보를 저장하는 0.01초주기 저장 데이터, 차량의 운행시간, 운행거리 등을 기록하는 1일 주기 저장되는 데이터로 구별됨.
- 특히 법적으로 1초주기 데이터는 6개월 이상의 정보를 단말기 내부에 저장하도록 하고 있으나 현재의 DTG 단말기가 데이터 위·변조에 취약하여 단말기의 내부 저장정보에 블록체인 프로토콜을 적용하여 위·변조 할 수 없도록 기술 개발을 수행하고자함.

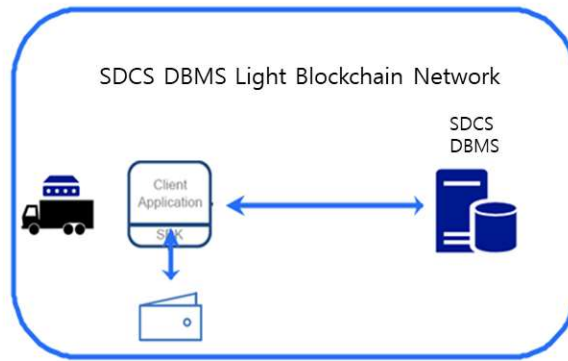


그림 4. SDCS DBMS

○ 2차 년도

(1) 카메라모듈 제작

- 차량에 설치되어 정지, 주행중에 전방의 도로 및 차량의 영상을 촬영함
- 고화질의 영상을 확보하기 위해 FHD카메라를 사용함
- 유선통신을 사용하여 촬영된 영상을 ADAS장치로 전송함

(2) ADAS 장치 제작(차량운행기록 장치 기능추가)

- 카메라모듈로부터 전송된 영상을 수신함
- 영상처리 기능을 사용하여 촬영된 영상으로부터 필요한 정보를 추출함
- 전방차량 추돌방지 기능 개발(처리속도 : 20fps, 정확도 : 95%)
- 차선이탈 경보기 기능 개발(처리속도 : 20fps, 정확도 : 95%)
- 급차로변경 확인 기능 개발(처리속도 : 20fps, 정확도 : 95%)
- 안전거리 미확보 확인 기능 개발(처리속도 : 20fps, 정확도 : 95%)

(3) SDCS(Safety Driving Compensation System) 안전운전 프라이빗 블록체인 네트워크 소프트웨어 개발

- SDCS 사고 안전 네트워크는 도로위에 안전운전 정보를 제공하기 위한 네트워크로서 도로상의 사고정보와 SDCS 데이터베이스로부터 분석된 위험운전 행동을 GPS위치기반 정보를 바탕으로 누적된 안전운전 정보를 제공하는 시스템
- 네트워크에 참여하는 기관이 도로상의 사고안전정보를 공유함으로써 교통사고를 예방하고자 하는 시스템으로 블록체인 기반의 공유원장 기술을 적용함으로써 신뢰성 높은 정보를 제공할 수 있는 블록체인 네트워크를 개발하고자 함.

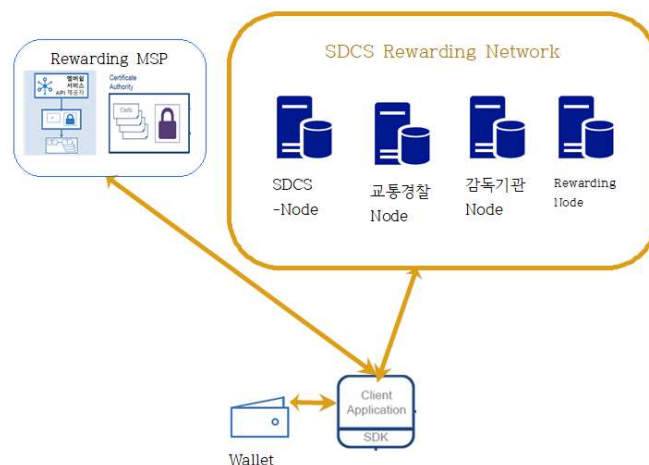


그림 5. SDCS Rewarding 블록체인 네트워크의 구성

(6) 초당 트랜잭션 처리 횟수 : 500TPS 이상

(7) DTG 단말기 고속도로 테스트 데이터 셋 구성

- 경부고속도로 양재->천안 도로 정보 테스트 데이터 셋 (77km)

○ 3차 년도

- (1) 시스템통합
 - 통신모듈과 서버연동(TCP 루프백테스트 진행)
 - 시험차량 및 도로환경에서 차량주행상태에 따른 차량운행기록장치의 시험진행
 - 차량주행시험시 발생한 문제점 분석 및 개선작업 진행
 - 차량운행기록장치와 차량관제용서버의 연동시험
 - 연동시험시 발생한 문제점 분석 및 개선작업 진행
- (2) ADAS 장치 제작(차량운행기록 장치 기능추가)
 - 규정차선 확인기능 개발(처리속도 : 20fps, 정확도 : 95%)
 - 중앙선침범 확인기능 개발(처리속도 : 20fps, 정확도 : 95%)
 - 불법유턴 확인기능 개발(처리속도 : 20fps, 정확도 : 95%)
- (3) 통신기능 개발
 - 무선통신(LTE) 모듈을 사용하여 차량운행기록장치와 서버간의 통신기능 개발
- (4) 차량관제 서버개발
 - 클라우드 서버를 이용하여 차량운행기록장치로부터 전송된 차량운행데이터 수신기능 개발
 - 서버 DB에서 차량운행데이터 저장 및 관리 기능 개발
 - 블록체인 서버와 통신을 위한 API 개발
 - 애플리케이션 서버와 통신을 위한 API 개발
- (5) 애플리케이션 서버 개발
 - 차량관제용서버 통신용 API 기능 개발
 - 안전운전 보상 마일리지 확인기능
 - 서버내에서 차량운행데이터 정상표시(속도, RPM, 브레이크신호, GPS) 유무확인 진행
- (6) 모바일 Dapp 개발
 - 블록체인서버 통신용 API 기능개발
 - 안전운전 보상 마일리지 확인기능
 - Dapp내에서 차량운행데이터 정상표시(속도, RPM, 브레이크신호, GPS) 유무확인 진행
- (7) SDCS(Safety Driving Compensation System) Rewarding 블록체인 네트워크 소프트웨어 개발
 - SDCS 보상 블록체인 네트워크는 SDCS 사고안전 네트워크에서 제공되는 도로상의 누적된 안전운전 정보를 운행하고 있는 차량에 제공함으로써 안전운전을 유도하기 위한 블록체인 네트워크임.
 - 즉, 도로상의 환경과 차량흐름을 바탕으로 도로상황에 적합한 안전운전 조건을 운전가이드를 제공하고 이러한 운전가이드를 차량이 수용함으로써 사고위험을 줄이고자하는 네트워크
 - 제안된 안전운전 가이드정보와 이를 수행한 차량 정보를 바탕으로 보상을 제공하고자 하는 네트워크가 SDCS 보상 블록체인 네트워크이며, 보상의 결과는 각각의 운행차량에 점수 나 마일리지 형태로 전자지갑에 기록되도록 하는 시스템을 개발하고자 함.

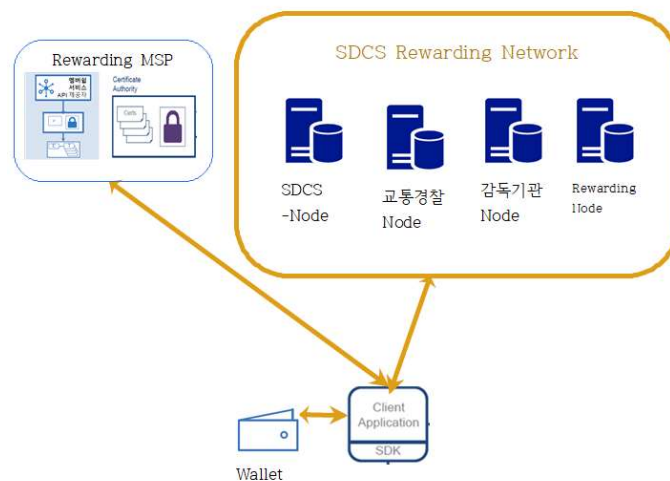


그림 6. SDCS Rewarding 블록체인 네트워크의 구성

- (8) 시스템 통합작업(SDCS 네트워크와 디지털 운행기록 단말기 통합)
 - DTG 블록체인 단말기 시제품을 네트워크에 통합하여 성능을 시험하는 단계로서 SDCS DBMS 구축, 사고안전 블록체인 네트워크, 보상네트워크가 유기적으로 운영되도록 시스템을 개발 보완함.(초단 트랜잭션 처리회수 : 500TPS이상)

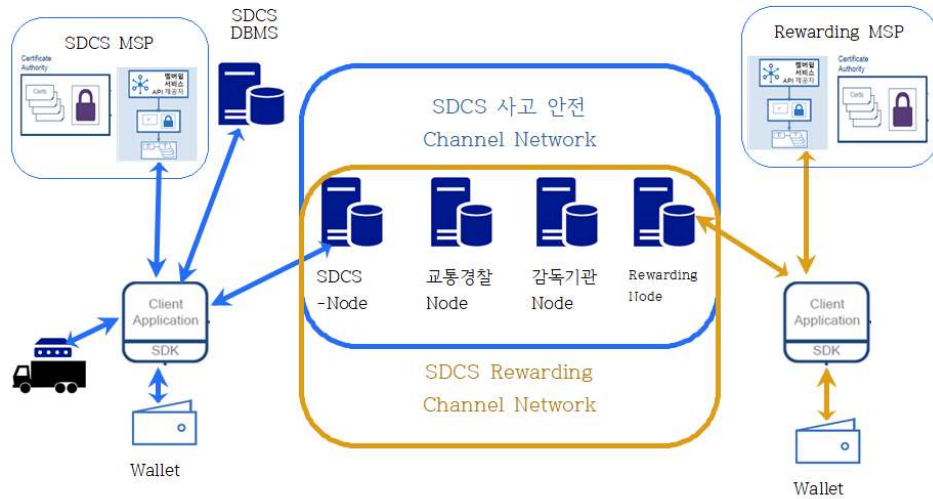


그림 7. DTG 단말용 블록체인 및 블록체인기반 안전운전 보상시스템 구성도

(9) DTG 단말기고속도로 보상정보 데이터 셋 구성

- 경부고속도로 양재->천안 보상운전 데이터 셋 (건/km 100건)

구분	연구개발목표	핵심 연구개발성과 (연구개발성과물)	현재 국내·외 기술수준	연구개발 목표치	목표치 설정 근거	가중치 (%)
1년차 (2019)	차량운행기록장치 제작	J1939/J1708 Dongle을 사용 한 차량운행데이터 추출	-	-	표준	6
		차량운행기록 메인장치 개발	-	-	표준	3
	블록체인보상시스 템개발	차량운행기록장치 라이트 블록체인	7~20TPS	500TPS	비트코인/이더리움	6
2년차 (2020)	ADAS장치제작	카메라모듈 제작	모빌아이	30FPS, 95%	업계 선두	2
		영상처리장치 HW 개발	모빌아이	30FPS, 95%	업계 선두	3
	블록체인보상시스 템개발	블록체인 서버 네트워크	7~20TPS	500TPS	비트코인/이더리움	6
3년차 (2021)	차량관제서버개발	차량운행데이터 저장용 DB생 성	200ms	-	표준	2
		서버 S/W와 DB연동	200ms	-	표준	2
	블록체인보상시스 템개발	블록체인 보상 네트워크	7~20TPS	500TPS	비트코인/이더리움	6

2. 연구개발과제의 수행 과정 및 수행 내용

○ 연구개발 추진전략·방법

1. 기술정보 수집

가. 디지털운행기록장치(Digital Tachograph) H/W, S/W 측면

- 1) 현재 국내 디지털운행기록장치 제작업체는 35개 이상으로 관련 정보 다수
- 2) 자사의 R&D능력으로 33개월 이내 기본적인 디지털운행기록장치 개발 가능(향후 고도화)
- 3) 당사가 개발하고자 하는 디지털운행기록장치의 제품 차별점은 ADAS장치를 통해 화물차 교통사고 주요 원인(차선이탈, 중앙선 침범, 급차로 변경, 불법유턴, 안전거리 미확보 등)을 확인하는 FHD카메라 영상처리 알고리즘의 개발은 당사가 현재 개발 중인 2018 국토부과제(사회이슈해결형)의 한양대학교 영상처리 기반기술이 확보되어 추가 개발 가능함.

나. V2X 블록체인 기반의 안전운행 보상 핵심 알고리즘 기술정보

- 1) 2018년 특허청 IP나래지원사업을 통해 기본적인 기술정보 분석 완료 함
- 2) 상기 과제 이후 지속적으로 특허법인과 협력관계를 통해 본 과제 관련 선행 특허 5건 확보
- 3) 현재 신청 완료된 2019 특허청 IP-R&D 사업을 통해 블록체인 알고리즘 핵심특허 도출예정

2. 전문가 확보 방안

다. 디지털운행기록장치(Digital Tachograph)

- 1) 관련업 전문가 추가 영입 및 자사의 R&D능력으로 33개월 이내 H/W, S/W 개발 가능함.

라. V2X 블록체인 기반의 안전운행 보상 알고리즘

- 1) 위탁기관인 상명대학교 산학협력단과 기술개발 가능
- 2) 당사 중기부과제의 일부 기술을 전자부품연구원의 기술이전을 받아 개발 중인 협력 관계가 있어 필요시에 전자부품연구원과 협업을 통해 기술협력 가능함.
- 3) 총괄책임자의 블록체인 전문가 풀을 이용하여 일부 기술 용역을 통해 해결
- 4) 당사 책임급 연구인력 충원을 통해 해결

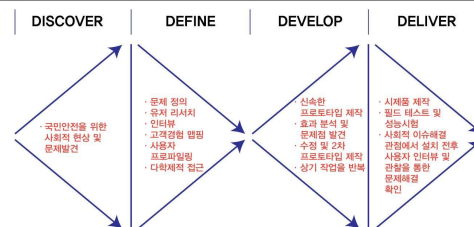
3. 연구개발 접근방법

일반 기술 중심의 R&D방법과 차별화된 당사의 서비스디자인 R&D접근방법

- ① 문제 발견 : 화물차 교통사고 원인에 대한 사용자 조사(언론 및 통계자료)
- ② 문제 정의 : 사용자 조사 결과에 대한 분석
- ③ 구체화 : 분석 결과를 적용한 시제품 설계
- ④ 프로토타입 제작 : 간단하고 빠른 시제품 제작 및 필드테스트를 통한 이론, 성능 검증

(사용자 인터뷰) > 다시 ①으로 회귀 진행

연구개발 추진전략 - 서비스 디자인적 접근을 통한 연구개발 추진전략 : 더블다이아몬드 모델



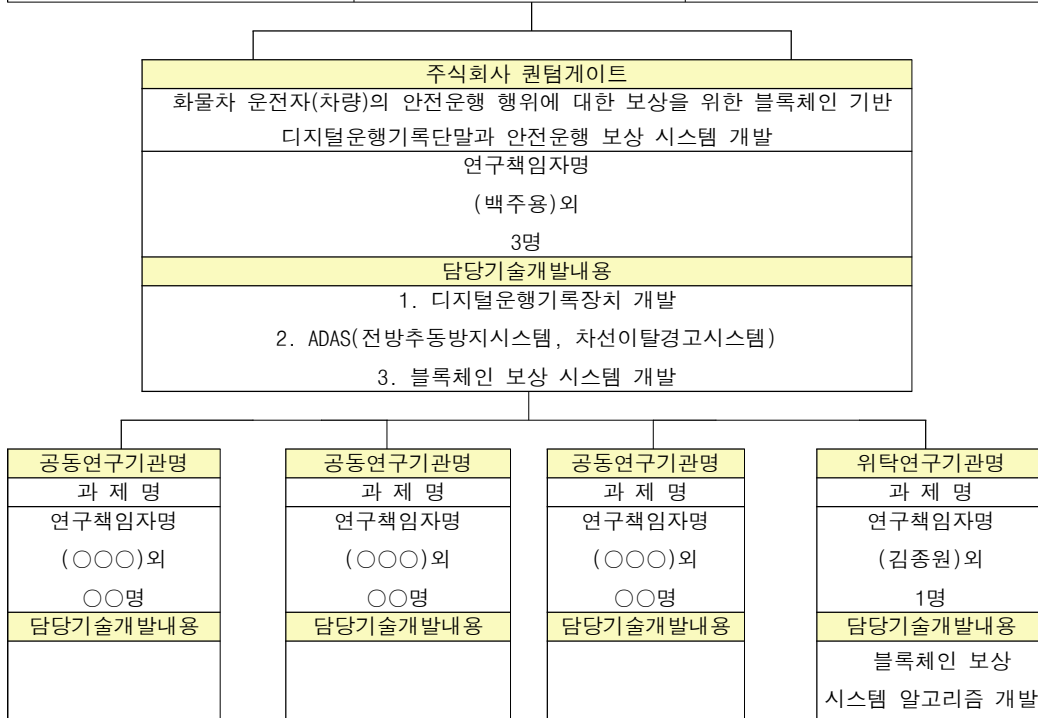
[서비스디자인 더블다이아몬드 모델]

○ 연구개발 추진체계

- 1) 내용 중 “원숫자”는 성과점검기준표의 질적성과지표 번호임
- 2) ①카메라영상수집기능, ②전방차량추돌방지기능 및 ③차선이탈경보기능 중 ②와③은 Movon社에서 ADAS 공인인증 관련 기 인증한 기능으로, 당사 개발 제품과 비교 우위임에 따라 한시적 적용 결정

연구개발과제		총 참여 연구원
과제명	화물차 운전자(차량)의 안전운행 행위에 대한 보상을 위한 블록체인 기반 디지털운행기록단말과 안전운행 보상 시스템 개발	주관연구책임자 (백주용)외 총 7명

기관별 참여 현황		
구분	연구기관수	참여연구원수
대기업		
중견기업		
중소기업	1	3
대학	1	2
국공립(연)		
출연(연)		
기타		



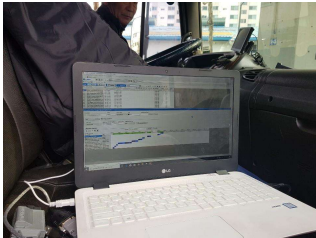
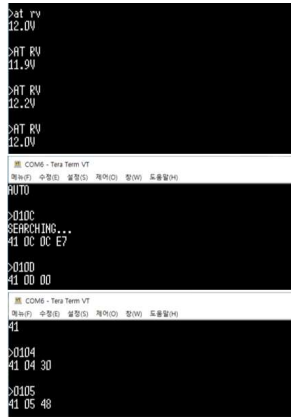
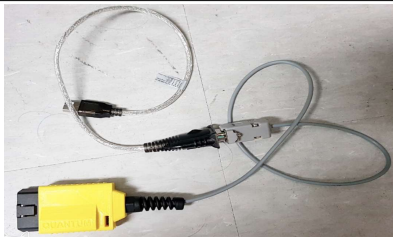
○ 연구개발 추진일정

1차년도															
일련 번호	연구내용	추진 일정												연구 개발비 (단위: 천원)	책임자 (소속 기관)
		1	2	3	4	5	6	7	8	9	10	11	12		
1	J1939/J1708 Dongle 제작													3000	김도원
2	차량운행기록 메인 장치 개발													20000	김도원
3	시험환경 데이터 수집													3000	김도원
4	J1939/J1708 장치 S/W 개발													10000	김도원
5	차량운행기록 메인 장치 S/W 개발													30000	김도원
6	CAN 통신 S/W 개발													10000	김도원
7	차량운행기록장치 라이트 블록체인 개발													30000	김도원
2차년도															
1	카메라 모듈제작													2000	김도원
2	영상처리장치 HW 개발													30000	김도원

3	시험환경 데이터 수집															3000	김도원
4	카메라영상 수집기능 개발															5000	김도원
5	전방차량 추돌기능 기능개발															20000	김도원
6	차선이탈 경보기능 개발															20000	김도원
7	규정차선 확인기능 개발															20000	김도원
8	중앙선 침범 확인기능 개발															20000	김도원
9	급차로변경 확인기능 개발															20000	김도원
10	불법유턴 확인기능 개발															20000	김도원
11	안전거리 미확보 확인기능 개발															20000	김도원
12	블록체인 서버 네트워크 개발															30000	김종원
3차년도																	
1	시제품제작															6000	김도원
2	차량운행데이터 저장용 DB 생성															1000	김도원
3	서버 S/W와 DB연동															10000	김도원
4	통신모듈과 서버연동															3000	김도원
5	차량운행기록장치 연동															10000	김도원
6	애플리케이션 서버 연동															20000	김도원
7	모바일 Dapp 연동															20000	김도원
8	유사환경하에서 동작확인															30000	김도원
9	블록체인 보상 네트워크															30000	김종원

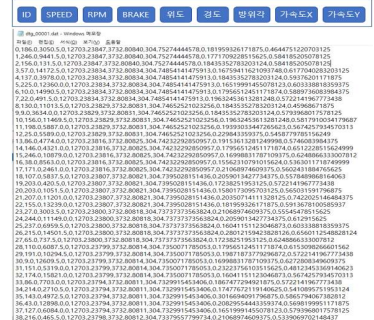
○ 연차별 연구개발 내용

1차년도 (2019년) 연구개발

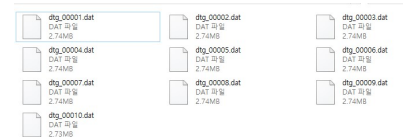
연구 범위	연구 수행 방법 (이론적 · 실험적 접근방법)	연구개발 수행 내용
차량 J1939/1708 시험환경구축	J1939/J1708 Dongle을 사용하여 차량데이터 확인(속도, rpm, 브레이크)	 실차 OBD2 포트  실제 차량테스트  차량전압, 전압, RPM
J1939/J1708 Dongle 제작	차량데이터 신호 추출장치 제작	 OBD2 Dongle 시제품
차량운행기록장치 제작	1. CAN통신장치를 사용하여 차량 데이터 수신 2. GPS신호 수신 3. 가속도센서를 이용하여 충격 감지 4. 데이터 저장장치 5. 디스플레이/입력장치 6. 오디오 장치	 차량운행기록 메인장치(시제품)

시험 환경

10일치 데이터 수집



차량운행기록장치 데이터형식



10일치 데이터 파일

```
./Lblock data/0000120191111.d 00001 build
./Lblock data/0000220191112.d 00002 build
./Lblock data/0000320191113.d 00003 build
./Lblock data/0000420191114.d 00004 build
./Lblock data/0000520191115.d 00005 build
./Lblock data/0000620191116.d 00006 build
./Lblock data/0000720191117.d 00007 build
./Lblock data/0000820191118.d 00008 build
./Lblock data/0000920191119.d 00009 build
./Lblock data/0001020191120.d 00010 build
./Lblock data/0001120191121.d 00011 build
./Lblock data/0001220191122.d 00012 build
./Lblock data/0001320191123.d 00013 build
./Lblock data/0001420191124.d 00014 build
```

라이트블록체인 인코딩

프로그램 실행 결과

```
143f3e0a4aa388e1ee519f55118f5be3400dd989 D | 20191121 04:49:44
19a51a08084dcf2eb064adbedf8f3b342441db5 X | 20191121 16:40:16
fd13071579a1f426fbad54f3498eb83b131930a5 V | 20191121 16:20:05
5688e0ec2e6701bb87c257b03783c4b9cb203ef0 V | 20191121 16:20:05
5683cf74db5c581a3677f43f9274fddbbeefaf30ec1 V | 20191121 16:20:05
99a626084e485e2e457ad639a2ff7fe77d8913d V | 20191121 16:20:05
7fe33f2692eb78ed0496a125080893168c2d434 V | 20191121 16:20:05
a30b0ee2baa5554c9f5ae338083f39770a399 V | 20191121 16:20:05
952e90151da7f27fb39a9ef7cb455116f9a01d V | 20191121 16:20:05
d7ee4e2f3dc0f02a6123182359d4567edc99c6 V | 20191121 16:20:05
296996380f9fb76e9482221513df8d896dff26a V | 20191121 16:20:05
cb9ce0b1c1e5f72ab553cfe7de54c3d3792d5c V | 20191121 16:20:05
11954c3e51f18e10085c608f9919a17511dc0 V | 20191121 16:20:05
372b13ac35a350ee9ec9ec47f1f302fec1ffafdd V | 20191121 16:20:05
7da0bc02e0bb6589bae64c55e6732b4cd91d6c V | 20191121 16:20:05
```

라이트블록체인 디코딩

프로그램 실행 결과(정상)

```
d6ead501c45ccc43f3e0a4aa388e1ee519f55118f5be3400dd989 D | 20191121 04:49:44
18f5be3400dd98919a51a08084dcf2eb064adbedf8f3b342441db5 X | 20191121 16:40:16
fd0f3b342441db513071579a1f426fbad54f3498eb83b131930a5 V | 20191121 16:20:05
98eb83b131930a55688e0ec2e6701bb87c257b03783c4b9cb203ef0 V | 20191121 16:20:05
783c4b9cb203ef05e83cf74db5c581a3677f43f9274fddbbeefaf30ec1 V | 20191121 16:20:05
fbd0d1eeefaf30ec1199a626084e485e2e457ad639a2ff7fe77d8913d V | 20191121 16:20:05
39627ff77d8913d76a33f2692eb78ed0496a125080893168c2d434 V | 20191121 16:20:05
080893168c2d434a30b0ee2baa5554c9f5ae338083f39770a399 V | 20191121 16:20:05
38883f39770a39952e90151da7f27fb39a9ef7cb455116f9a01d V | 20191121 16:20:05
cb455116f9a01dd7ee4e2f3dc0f02a6123182359d4567edc99c6 V | 20191121 16:20:05
952e90151da7f27fb39a9ef7cb455116f9a01d V | 20191121 16:20:05
13df8d896dff26acb9ce0b1c1e5f72ab553cfe7de54c3d3792d5c V | 20191121 16:20:05
ed54c3d3792d5c11954c3e51f18e10085c608f9919a17511dc0 V | 20191121 16:20:05
0f9919a17511dc0372b13ac35a350ee9ec9ec47f1f302fec1ffafdd V | 20191121 16:20:05
1f302fec1ffafdd7da0bc02e0bb6589bae64c55e6732b4cd91d6c V | 20191121 16:40:16
00732b4cd91d6c40272f1c5a0e702f5a0bdc7ba0d705c3f5af V | 20191121 16:40:16
4c7ba0d705c3f5af569692324b23e761275e1a18f0eb38d30d8d4 V | 20191121 16:40:16
dfe58b38d30d8d4c8f2ad30b350c71f103cdae3a33ce95f47 V | 20191121 16:40:16
30a313ce95f47d0b9684c5c40e623f0977400d85326eb0220b V | 20191121 16:40:16
9d85326eb0220b9c80668243c80602377da13b35835de0e74139 V | 20191121 16:40:16
330b35835de0e74139279ac4c3aa79c1551d90d308ce427abddff V | 20191121 16:40:16
390ccee427abddff92364180adceb09b152bea37b79c65e56f6a9ee V | 20191121 16:40:16
79c50e56f6a9ee7156801861e08c03a43b329aac08a2ebdd390c V | 20191121 16:40:16
c08a2ebdd390c8f1f5a0e1f0e1c7910ab072c302c4f5980191 V | 20191121 16:40:16
c502c4f59801919303c089d41e0d18c47914b085e0f908eba108dc50 V | 20191121 16:40:16
0f908eba108dc50c4c0260351899f033152239b0eb0c2440c74 V | 20191121 16:40:16
9b0eb0c2440c74c1becb89f4009cc85321e0219ca72a227066ea V | 20191121 16:40:16
a072a227066ea5e025933ca0414ed12081a508140977a0f5a7a V | 20191121 16:40:16
140977a0f5a7a31c201ee0edf2781708e42918f4c0d96132 V | 20191121 16:40:16
918f4c0d96132a3a024997565084c0bb7f1ac9023b7149933fff V | 20191121 16:40:16
9023b7149933fff1fda21a9052603fca522e4aa30d117f068790ba V | 20191121 16:40:16
```

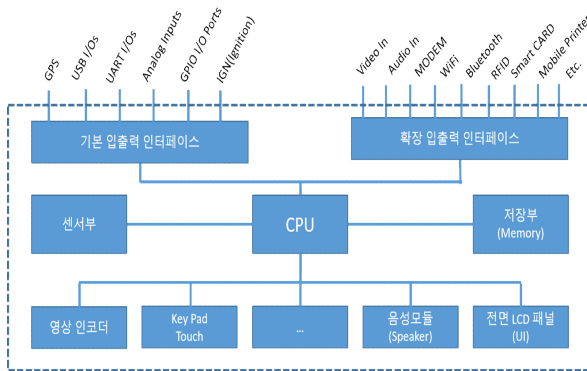
라이트블록체인 디코딩

프로그램 실행 결과(변조)

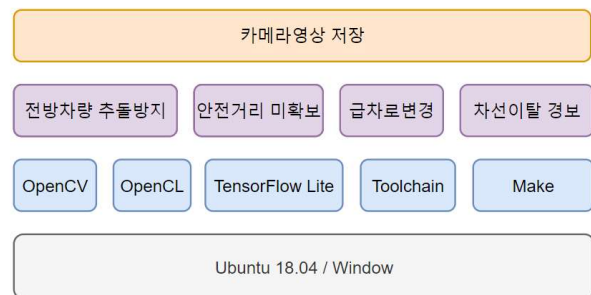
1. 영상처리 알고리즘 개발

아래의 그림은 1차연도 개발 결과물인 DTG기능과 2차연도 개발 결과물인 ADAS기능을 포함하는 S/W Stack임. 1차연도 개발 결과물인 DTG S/W 기능이 이미 본 HDK에 이식된 상태이며, 상세 설명은 생략함. 윈도우시스템에서 영상처리 알고리즘을 개발·검증한 후 Toolchain, Make 등을 활용하여 리눅스 환경으로 포팅 진행함.

카메라 비전 라이브러리인 OpenCV 및 딥러닝 모델을 활용하여 ①카메라영상수집기능, ②전방차량추돌방지기능, ③차선이탈경보기능, ⑥급차로변경확인기능, ⑧안전거리미확보확인기능을 개발²⁾ 하였으며 영상처리 알고리즘을 최적화 하기 위해 OpenCL, TensorFlow Lite 등의 라이브러리를 활용하여 개발 진행함.



Architecture of Digital Tachograph



ADAS Simulation S/W Stack with DTG

① 알고리즘 시험용 시뮬레이터

개발된 알고리즘의 기능 학인을 위한 시험은 아래의 그림과 같이 FHD 이상의 화질을 보이는 모니터를 이용하여, 이미 취득된 영상을 플레이한 후, ADAS Camera를 LOS(line of sight)상에 배치하여 실시간으로 모니터상의 영상을 취득하면서 차량의 주행 상태를 기능에 맞게 감지·수행하는지를 테스트베드를 구성하였음. 단, 이 테스트베드는 비디오소스 컴퓨터에서 표출하는 영상으로는 ②전방차량추돌방지기능과 ③차선이탈경보기능만 시뮬레이션이 가능함.

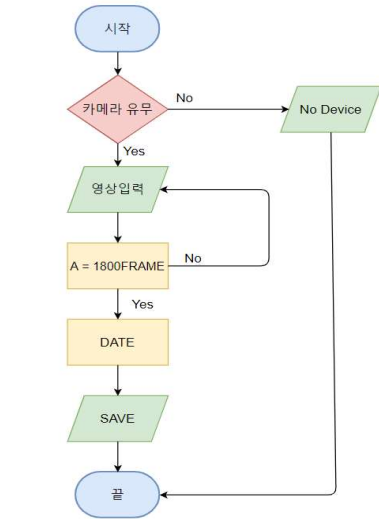


[그림] ADAS 기능 시뮬레이션 테스트베드

② 카메라 영상 수집기능 개발 <질적목표 No.1>

아래의 좌측 그림은 카메라 영상 수집기능 알고리즘의 플로우차트로서 카메라 영상을 30FPS기준 1800Frame마다 저장하여 1분 단위의 주행영상을 수집하였으며 크로스컴파일을 이용하여 임베디드 리눅스 환경에서 실행하였음. 우측 그림은 실행 영상 저장파일로서 1분단위의 화물 상용차 주행영상을 수집한 결

과디렉터리를 캡처함.



영상 수집기능 알고리즘

	2020-10-20_09h58.wmv 길이: 00:01:00	프레임 높이: 480 프레임 너비: 640
	2020-10-20_09h59.wmv 길이: 00:01:00	프레임 높이: 480 프레임 너비: 640
	2020-10-20_10h00.wmv 길이: 00:01:00	프레임 높이: 480 프레임 너비: 640
	2020-10-20_10h01.wmv 길이: 00:01:00	프레임 높이: 480 프레임 너비: 640
	2020-10-20_10h02.wmv 길이: 00:01:00	프레임 높이: 480 프레임 너비: 640
	2020-10-20_10h03.wmv 길이: 00:01:00	프레임 높이: 480 프레임 너비: 640
	2020-10-20_10h04.wmv 길이: 00:01:00	프레임 높이: 480 프레임 너비: 640

프로그램 실행 영상저장파일

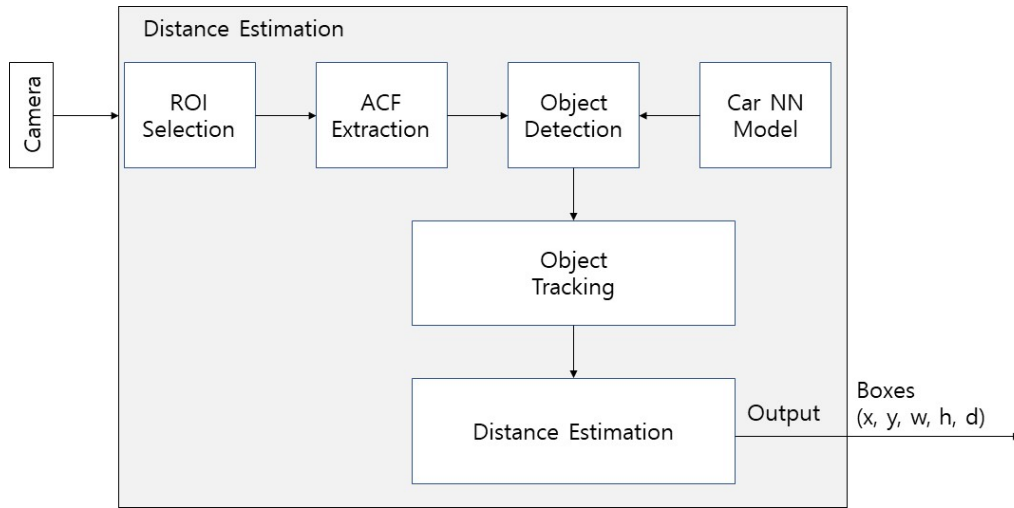
- 개발검증(방법/기준/결과)

* 상용 제품 기준 동영상 저장 기능이 확인되어 적합 판정

CASE	검증방법	검증기준	검증결과
카메라 영상 수집기능	- 시스템 작동 시 카메라 영상 저장파일 확인 10분동안 실행 시 10개의 파일이 저장	1분마다 동영상 저장 - 동영상 저장된 시간파일명 생성확인	2020-10-20_09h58.wmv 길이: 00:01:00 프레임 높이: 480 프레임 너비: 640 2020-10-20_09h59.wmv 길이: 00:01:00 프레임 높이: 480 프레임 너비: 640 2020-10-20_10h00.wmv 길이: 00:01:00 프레임 높이: 480 프레임 너비: 640 2020-10-20_10h01.wmv 길이: 00:01:00 프레임 높이: 480 프레임 너비: 640 2020-10-20_10h02.wmv 길이: 00:01:00 프레임 높이: 480 프레임 너비: 640 2020-10-20_10h03.wmv 길이: 00:01:00 프레임 높이: 480 프레임 너비: 640 2020-10-20_10h04.wmv 길이: 00:01:00 프레임 높이: 480 프레임 너비: 640

③ 전방차량 추돌방지기능 개발 <질적목표 No.2>

전방 카메라 영상이미지에 관심영역인 ROI(Region of Interest)를 설정하여 관측 범위를 지정하고, ACF Extraction(Extraction of Aggregated Channel Features)을 통해 이미지의 고유 특징을 추출함. ACF로 추출된 Gradient magnitude, LUV color channels 및 quantized gradient angle 기반의 Object Detection을 통해 차량을 감지하고 Vehicle Tracking을 수행. Distance estimation을 위해 카메라 고유의 파라미터를 추출하고, Distortion, 렌즈 왜곡 등을 제외한 좌표계상의 Coordination을 Output으로 출력함. 출력된 좌표 데이터가 안전거리 이내에 진입하는지 여부를 판단하여 전방차량 추돌방지 기능을 수행함



2019년 현재 FCA(Forward Collision Avoidance Assist, 전방충돌방지시스템) 제동 기준 시간

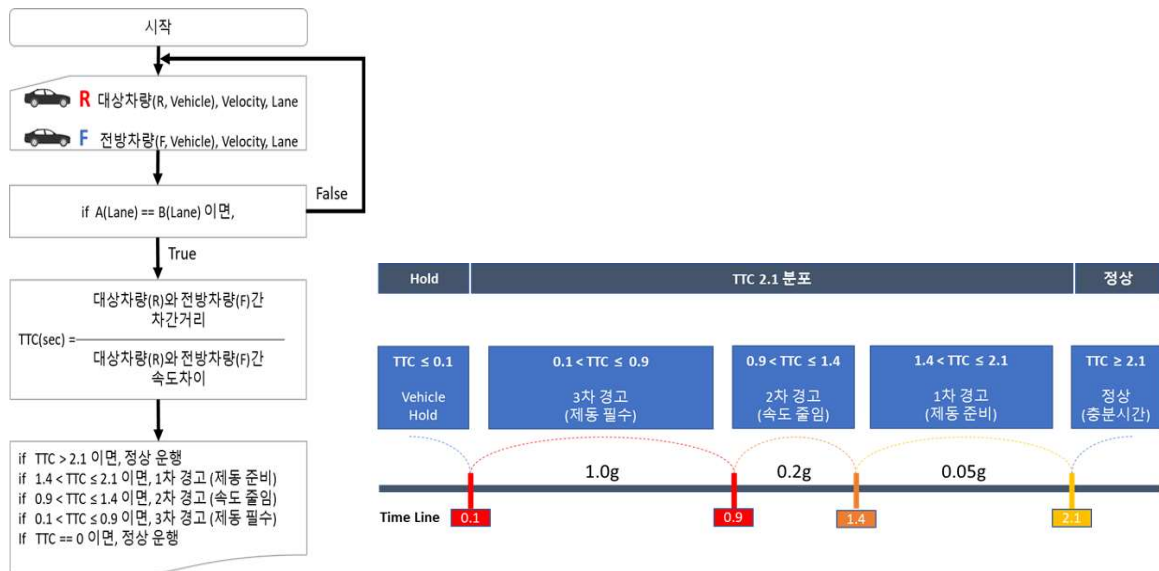
- ① 전방 충돌 예상시간 약 2.1초일 때, 1차 경고 (제동 준비)
- ② 전방 충돌 예상시간 약 1.4초일 때, 2차 경고 (0.2g 제동)
- ③ 전방 충돌 예상시간 약 0.9초일 때, 3차 경고 (1.0g 제동)
- ④ 2초간 자동 정차 후 2초 후 해제

운전자 전방차량 인지 후 제동가능 시간

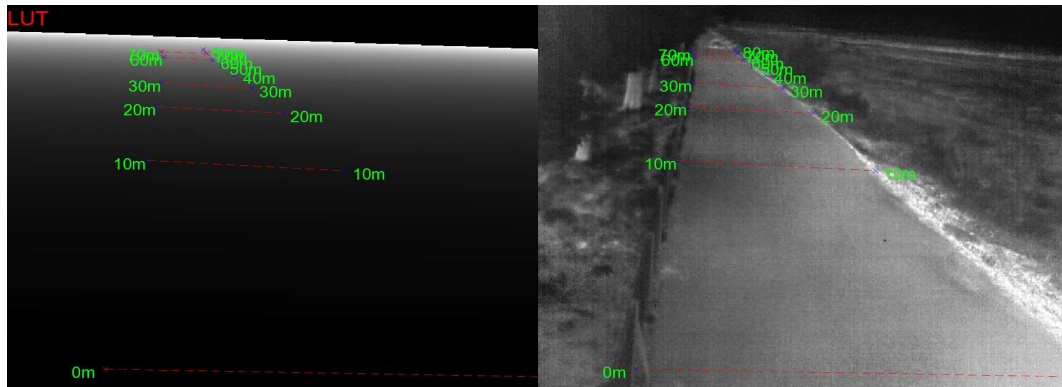
- ① 운전자 0.5~0.7초 시간 필요
- ② 시속 100Km/h 일 때, 시간별 주행거리 : 0.5초 - 약 13.9m, 1초 - 약 27.78m

따라서, 차량 시속 80Km/h일 때, 앞차와의 거리가 80m 이상 되어야 안전하게 정차할 수 있음.

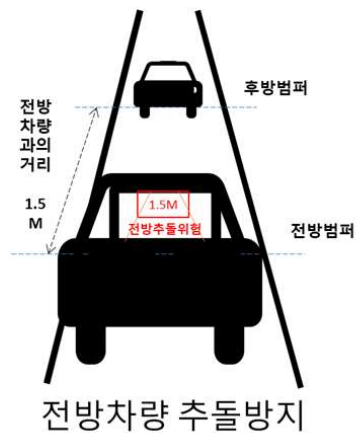
FCA(Forward Collision Assist, 전방충돌방지 시스템)에 사용하는 전방충돌예상 시간을 기준으로 1차, 2차, 3차 경고로 나누어 구성함. 아래는 전방 충돌예상시간 알고리즘임.



전방 차량과의 거리 및 차량의 속도 추정을 위한 영상좌표-실제위치 LUT 생성 및 기준점



[그림] 영상좌표-실제 위치 LUT 생성 결과 및 기준점 표시



전방추돌경고(FCW) 설정	
FCW 기능	On/Off
민감도	LV1 - LV5
고속 민감도	On/Off
경고 반복	On/Off
브레이크 연동	On/Off
경고음	진동, S1 - S5

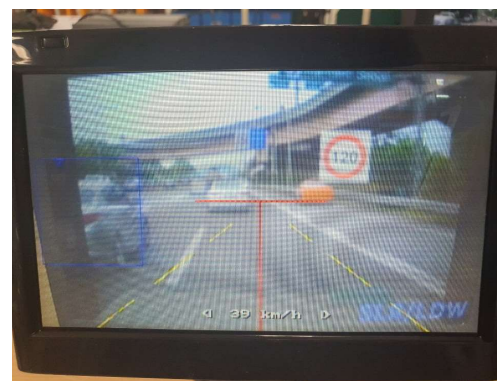
- 전방 차량과의 거리 추정 방법 연구
- 차량의 속도 추정 방법 연구
- 전방 차량과의 거리 및 차량의 속도 추정 기술 접목

시험(TTC 반영)

- 차량 속도: 전방차량 - 76.73Km/h, 후방차량 - 76.87Km/h
- 앞·뒤차간의 거리: 17m



[그림] 전방차량추돌방지 테스트1



[그림] 전방차량추돌방지 테스트2



[그림] 전방차량추돌방지 테스트3

[그림] 전방차량추돌방지 테스트4

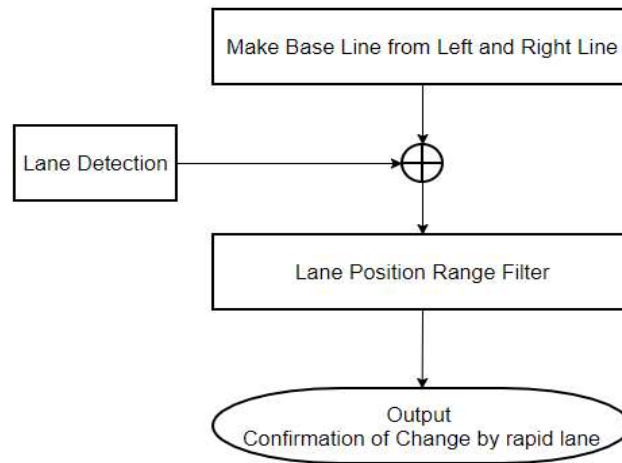
- 개발검증 결과

- * 국토교통부 전방차량추돌방지기능 규격 참조
- * 차량 FCA(Forward Collision Avoidance Assist, 전방충돌방지시스템)에서 사용하는 TTC(Time To Collision)기준을 적용하여 진행
- * 차량 시험 운행테스트에서 정확한 속도 및 거리 산출에 어려움 있음.
- * 차량 안전 운행상의 이유로 가변 속도 테스트에 어려움 많음.

CASE	검증방법	검증기준	검증결과		
목표자동차 정지 상태	● 운행자동차 속도 72km/h, 목표자동차로부터 150m 거리에 접근하였을 때 경고 ● 7회 중 5회 이상	● 전방추돌 경고 ● TTC 2.1초 이전에 충돌 경고	No	경고발생	TTC
			1	pass	1.8
			2	pass	1.9
			3	pass	1.7
			4	pass	1.8
			5	pass	1.7
			6	pass	1.9
			7	pass	1.8
목표자동차 감속운행	● 두 차량을 차간거리를 30~50m로 유지하며 운행, 72km/h의 초기속도로 주행 ● 목표자동차 서서히 감속 운행, 두 차량의 차간거리가 가까워짐에 따라 경고 ● 7회 중 5회 이상	● 전방추돌 경고 ● TTC 2.1초 이전에 충돌 경고	No	경고발생	TTC
			1	pass	1.8
			2	pass	2.1
			3	pass	2.0
			4	pass	1.7
			5	pass	1.9
			6	pass	1.0
			7	pass	1.4
목표자동차 저속주행	● 운행자동차 속도 72km/h로 150m 이상 간격을 두고 운행, 목표자동차 목표자동차는 30km/h의 일정 속도로 주행, 차간 거리가 가까워짐에 따라 경고 ● 7회 중 5회 이상	● 전방추돌 경고 ● TTC 2.1초 이전에 충돌 경고	No	경고발생	TTC
			1	pass	1.9
			2	pass	1.7
			3	pass	2.0
			4	pass	1.6
			5	pass	1.3
			6	pass	1.7
			7	pass	1.6

④ 차선이탈 경보기능 개발 <질적목표 No.3>

아래 그림은 차선이탈경보기능의 플로우차트로서 차선 인식 알고리즘을 이용하여 차선을 감지하고 기준 차선에서 좌우 차선과 기준선의 편차를 이용하여 차선이탈 확인기능을 수행함.



[그림] 차선이탈 경보기능 플로우 차트

아래 그림은 주행 차량에서 차선을 감지하고 차선과의 편차를 계산하는 영상임.



[그림] 차선 감지 기능1

[그림] 차선 감지 기능2

[그림] 차선 감지 기능3

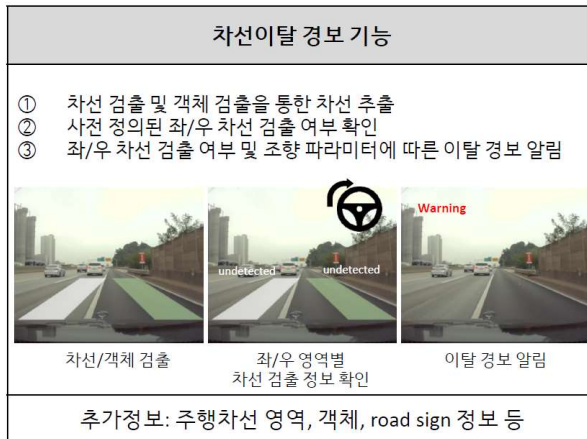
- 개발검증(방법/기준/결과)

- * 국토교통부 차선이탈경고 규격에 따라 자체시험을 진행하였음
- * 청각 / 시각 경고를 표출하면 Pass로 인정하여 도출된 결과임

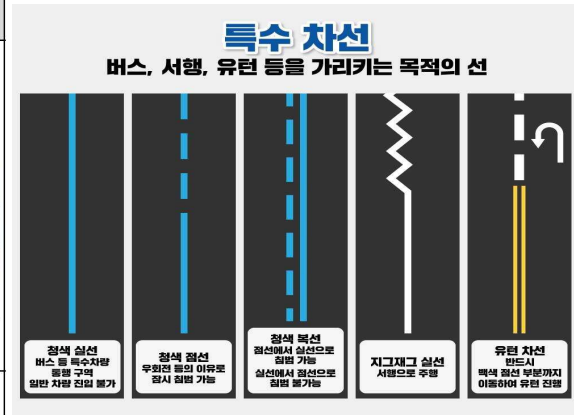
CASE	검증방법	검증기준	검증결과																																	
차선이탈경고	● 시속 65km/h 유지시키며 허용오차는 ±3km/h ● 임의의 조향핸들각을 입력하여 점진적으로 0.1m/s 내지 0.8m/s사이의 이탈속도로 좌측 또는 우측으로 주행 중인 차로를 벗어남 ● 다른 이탈속도로 시험 ● 앞서 시행한 방향과 반대인 방향으로 시험 ● 시험 총 횟수의 90%이상 검증기준을 만족	● 시각, 청각 또는 촉각 경고 중에서 2개 이상의 수단 ● 자동차가 차로를 이탈하는 방향을 운전자가 알 수 있도록 하는 촉각 또는 청각 경고 중 어느 하나의 수단 ● 차선에 가장 가까운 시험자동차의 전륜타이어 접지면 외측이 이탈되는 방향의 차선 바깥쪽 모서리에서 0.3m 떨어진 가상의 선 이내에서 차로이탈 경고를 제공	<table><tr><th>No</th><th>Buzzer</th><th>Indi.</th></tr><tr><td>1</td><td>pass</td><td>pass</td></tr><tr><td>2</td><td>pass</td><td>pass</td></tr><tr><td>3</td><td>pass</td><td>pass</td></tr><tr><td>4</td><td>pass</td><td>pass</td></tr><tr><td>5</td><td>pass</td><td>pass</td></tr><tr><td>6</td><td>pass</td><td>pass</td></tr><tr><td>7</td><td>pass</td><td>pass</td></tr><tr><td>8</td><td>pass</td><td>pass</td></tr><tr><td>9</td><td>pass</td><td>pass</td></tr><tr><td>10</td><td>pass</td><td>pass</td></tr></table>	No	Buzzer	Indi.	1	pass	pass	2	pass	pass	3	pass	pass	4	pass	pass	5	pass	pass	6	pass	pass	7	pass	pass	8	pass	pass	9	pass	pass	10	pass	pass
			No	Buzzer	Indi.																															
			1	pass	pass																															
			2	pass	pass																															
			3	pass	pass																															
			4	pass	pass																															
			5	pass	pass																															
			6	pass	pass																															
			7	pass	pass																															
			8	pass	pass																															
9	pass	pass																																		
10	pass	pass																																		
* Buzzer 및 Indicator로 경고																																				

⑤ 급차로 변경확인 기능개발 <질적목표 No.4>

차량 주행의 카메라영상에서 좌/우 차선을 검출하고, 조향 파라미터를 추출하는 기능을 구현함. 영상데이터를 수집하여 개발을 진행하였으며, 프로그램 알고리즘 수행을 위한 테스트 진행.

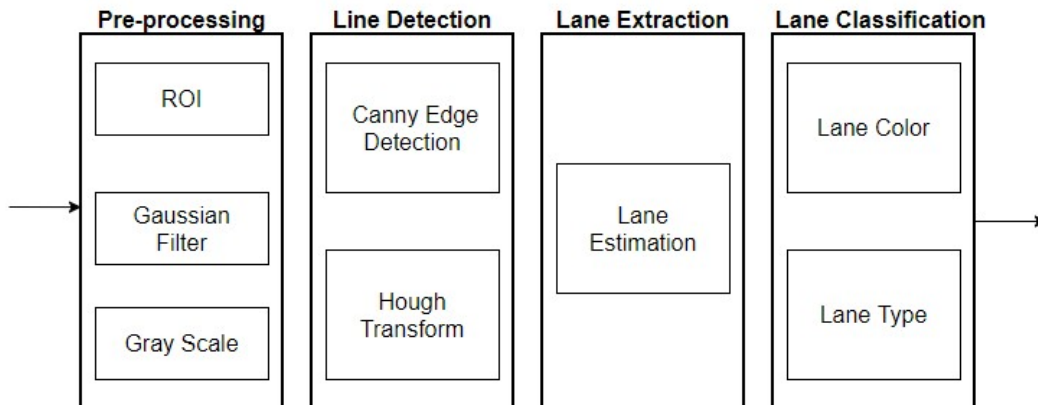


[그림] 차선이탈 경보 시나리오



[그림] 차선 정보

차선 검출 알고리즘을 수행하기 위한 차선검출 플로우차트에서 Computing Cost를 감소하기 위해 획득되는 영상에서 전방차선에 대한 ROI를 설정 후 Gaussian Filter 및 Gray Scale 작업을 통해 이미지 전처리 작업을 수행 후 Canny Edge Detection과 Hough Transform 알고리즘을 수행하여 해당 영상에서 차선을 추출하였음.



[그림] Lane Detection Flow Chart

차선 검출 알고리즘을 주행영상에 적용하였을 때 출력되는 이미지로서 Edge를 검출하고 Hough Transform 기반 차선 영역 검출 및 색상기반 차선 검출을 통해 차선 정보를 추출하는 과정을 나타냄.



[그림] Lane Detection 알고리즘 적용사진

카메라 영상 수집 기능 개발을 통해 수집한 주행영상으로 차선을 감지 할 수 있도록 기능 구현한 내용이며, 차선을 감지하면 초록색 선으로 표출함. 주/야간 주행 영상에서 차선 검출의 동일한 성능을 보임.



[그림] 동영상파일 입력 주간 야간 차선 검출 실험

본 시스템을 주행차량에 탑재하여 급차로 변경 알고리즘을 통해 차선 변경을 감지하는 시험으로 1차적으로는 인식된 차선과 기준선의 조향각(SAS-Steering Angle Sensor) 편차를 통해 차선의 변경을 감지하고, 2차적으로는 TTC 3.0초 이전에 차선 변경을 하면 급차로 변경으로 간주하고 감지하여 경고를 표시함.

- 차량 속도에 따른 차선변경 안전거리 표시
- 앞차와의 거리(정지거리)에 따른 차선변경 안전거리 표시
- 방향지시등연계에 따른 차선 변경 여부
- 좌측/우측 방향 센서 민감도 설정
- TTC 3초 이내에서 차선 변경시 급차로변경에 해당



급차로변경 설정	
급차로변경 경고	On/Off
전방근접 설정	2 - 20 미터 (2, 3, 5, 10, 20M)
좌측 민감도	LV1 - LV5
우측 민감도	LV1 - LV5
경고 반복	On/Off
경고음	Type1 - Type3

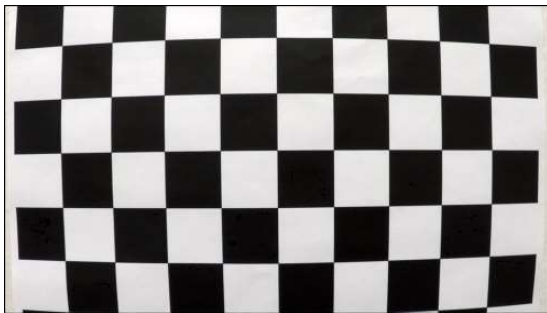
시험 테스트(TTC 반영 시험)

- 차량 속도: 전방차량 - 76.73Km/h, 후방차량 - 76.87Km/h
- 앞·뒤차간의 거리: 17m

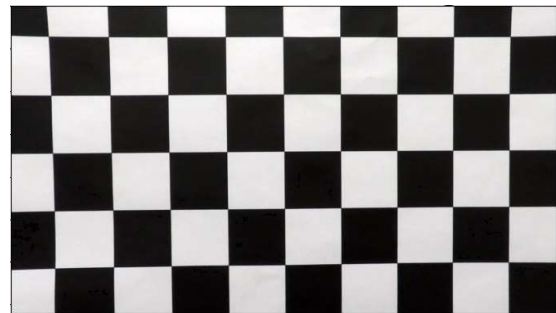
CASE	검증방법	검증기준	검증결과		
급차로 변경	● 시속 65km/h 유지시키며 허용오차는 ±3km/h ● 임의의 조향핸들각을 입력하여 점진적으로 0.9m/s 내지 1.6m/s사이의 이탈속도로 좌측 또는 우측으로 주행 중인 차로를 벗어남 ● TTC 3.0초를 확인 ● 앞서 시행한 방향과 반대인 방향으로 시험 ● 10회 중 90%이상 경고	● 갑작스런 차량 조향 핸들각 발생 확인 ● TTC 3.0초 이내 발생은 급차로 변경으로 간주 ● 차로이탈 경고	No	Buzzer	Indi .
			1	pass	pass
			2	pass	pass
			3	pass	pass
			4	pass	pass
			5	pass	pass
			6	pass	pass
			7	pass	pass
			8	fail	fail
			9	pass	pass
			10	pass	pass
			* 경고 *		

⑥ 안전거리 미확보 확인 기능개발 <질적목표 No.5>

이미지 프로세싱 성능확보를 위해 카메라 영상 캘리브레이션을 수행. 7x10 체커 보드의 코너를 검출하고 카메라의 내부 파라미터 및 왜곡 모델을 검출하고 모델을 이용하여 카메라 왜곡 보정을 수행하였음. 아래 그림은 왜곡 현상을 보정한 결과 이미지임.

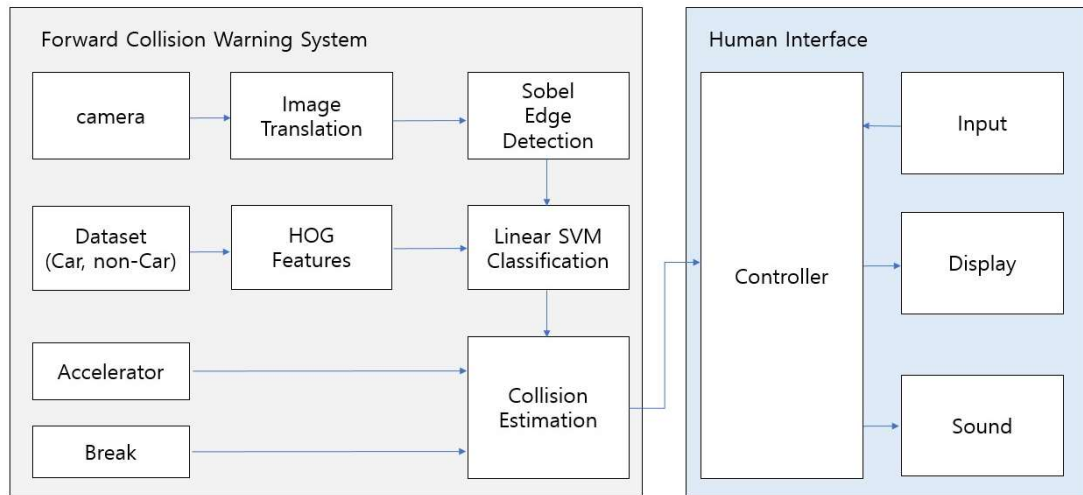


[그림] 카메라 왜곡 보정 전



[그림] 카메라 왜곡 보정 후

전방차량 추돌방지 기능의 블록다이어그램으로서 차량에 대한 오픈 데이터셋에서 HOG를 통해 특징을 추출하고 선형 SVM분류기를 통해 전방 카메라로부터 취득한 영상에서 차량을 검출하였음. 검출차량 오브젝트를 이용하여 Camera Geometry상의 전방 차량과의 거리를 산출하고 주행 차량으로부터 얻은 OBD 데이터와 전방 차량과의 거리 산출데이터를 이용하여 전방 추돌 예상시간을 산출하였음.



[그림] 전방차량 추돌방지 블록도

차량 및 비 차량으로 구분된 오픈 데이터셋을 이용하여 차량 학습 모델을 구현함. 차량 이미지 8792장과 비차량 이미지 8968장을 이용하여 학습을 수행하였음. 아래 그림은 학습을 수행한 차량 및 비차량 이미지임.



[그림] Car 이미지



[그림] non-Car 이미지

이미지 연산속도를 높이기 위해 이미지 사이즈를 재조정 하고 YCrCb 이미지로 변환하였음. 아래 그림은 원본이미지에서 YCrCb 적용 및 32x32로 조정한 결과임.

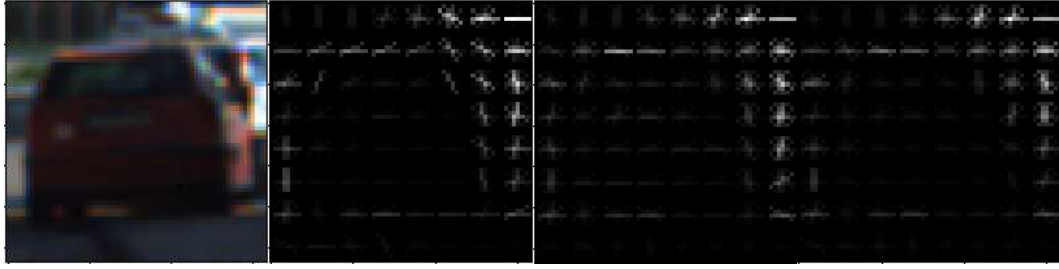


[그림] 원본 이미지



[그림] 조정 이미지

HOG(Histogram of Oriented Gradients) 알고리즘을 이용하여 이미지의 특징을 추출하고 SVC(Support Vector Classifier) 학습에 이용하였음. 아래 그림 15~17.는 차량 이미지에서 차량의 HOG 이미지를 추출한 결과임.



[그림] 차량이미지

[그림] Y HOG

[그림] Cr HOG

[그림] Cb HOG

카메라 FOV내의 모든 영역을 검색할 필요가 없으므로 카메라 뷰에 따른 차량 Box를 추적할 범위를 지정하였음. 아래 그림은 차량을 검색하는 모든 구간에 박스를 생성하여 ROI를 파악하기 위한 예시임.



[그림] 차량 검색 ROI 설정

전방 차량과의 거리 추정 방법 연구

- 두 차량 사이의 거리를 추정하기 위해 머신러닝 기반의 차량 검출 수행 결과와 영상좌표-실제위치 LUT를 사용하여 두 차량의 실제 위치를 추정한 후 추정한 두 위치의 차이로 전방 차량과의 거리 추정 값을 획득함.
- 카메라와 가까운 차량은 차량 검출 결과로 획득한 bounding box 좌상단 꼭지점에서부터 너비와 높이의 $\frac{1}{6}$ 만큼 이동한 좌표로 실제 위치를 추정하였고, 카메라와 먼 차량은 bounding box 아래쪽 경계선의 중심 좌표로 실제 위치를 추정함.



[도로환경에서의 전방 차량과의 거리 추정]

차량의 속도 추정 방법 연구

- 차량의 속도를 추정하기 위해 머신러닝 기반의 차량 검출 수행 결과와 영상좌표-실제위치 LUT를 사용하여 주행 차량의 실제 위치를 추정한 후 구간 별 실제 위치 변화를 기반으로 차량 속도 추정 값을 획득함.
- 차량 검출 결과로 획득한 bounding box 아래쪽 경계선의 중심 좌표로 실제 위치를 추정함.

$$P = LUT\left(B_x + \frac{B_w}{2}, B_y + B_h\right)$$

- 이때 P 는 차량의 추정 위치이고, B_x, B_y, B_w, B_h 는 bounding box의 x, y, width, height이다
- 전체 주행 구간을 아홉 구간으로 균등하게 분할한 후 세 구간마다 추정한 차량의 위치를 이용하여 구간 내 평균 속도를 추정함.



[그림] 차량의 속도 및 차간거리 표시

전방 차량과의 거리 및 차량의 속도 추정 기술 접목

- 레이저 거리 측정기(Leica Disto D510)를 사용하여 카메라로부터 가까운 차량의 뒷범퍼까지의 거리와 카메라로부터 먼 차량의 앞범퍼까지의 거리를 측정하였고, 두 거리의 차이로 실제 전방 차량과의 거리 값을 획득함.



[그림] 전방 차량과의 거리 추정 실험용 DB 획득 시 실제 거리 값 측정 방법 및 측정 도구 (Leica Disto D510)

시험 테스트(TTC 반영)

- 차량 속도: 전방차량 - 76.73Km/h, 후방차량 - 76.87Km/h
- 앞·뒤차간의 거리: 17m

```
[TS] ID 14 : BBox <501 194 141 60> / POS -0.00m / SPEED -76.73km/h / DIST 0.00m / TTC nans
[TS] ID 15 : BBox <59 232 60 37> / POS -55.57m / SPEED -76.87km/h / DIST 16.67m / TTC 116.1s
[VT]::DrawResults() 0...m_tracks.size() = 2
[IM]::DrawResults(img) ...

[IM] i = 0
[[[[[ Vehicles Info. ]]]]]

[IM] Draw ID: 14 / LANE: 1 / POS: 0m / SPEED: 77km/h / DIST: 0m / TTC: nans

[IM] i = 1
[[[[[ Vehicles Info. ]]]]]

[IM] Draw ID: 15 / LANE: 1 / POS: -56m / SPEED: 77km/h / DIST: 17m / TTC: 116.1s
```

[필드 테스트]

실제 주행차량에서 안전거리 미확보 확인기능을 테스트하였으며, 차량인식을 통해 검출한 전방 차량과의 거리가 일정 거리 이하로 진입하면 안전거리 미확보 상황을 인지하도록 설계.



[그림] 안전거리 미확보1



[그림] 안전거리 확보2



[그림] 안전거리 확보3



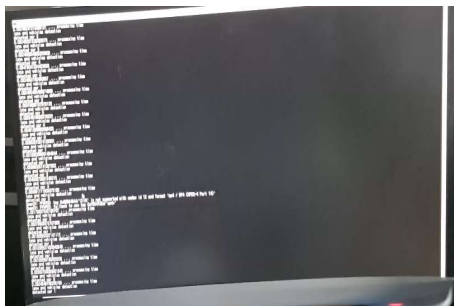
[그림] 안전 거리 확보4



[그림] 추돌예상시간 1.0초



[그림] 추돌예상시간 0.8초



[그림] 코드실행 화면



[그림] 전방추돌방지

- 개발검증 결과

- * 국토교통부 전방차량추돌방지기능 규격 참조
- * 안전거리 기준(일반)
 - 시속 60km/h 이상인 경우, 차량속도 거리비례
예) 60km/h일 때 안전거리 = 60m
 - 시속 60km/h 이하인 경우, 차량 속도 - 15

예) 45km/h일 때 안전거리 = 30m

- * 안전을 고려하여 45km/h의 저속 주행으로 시험 운행
- * 거리측정 결과 30m ±5m로 전방추돌경고 발생 검증

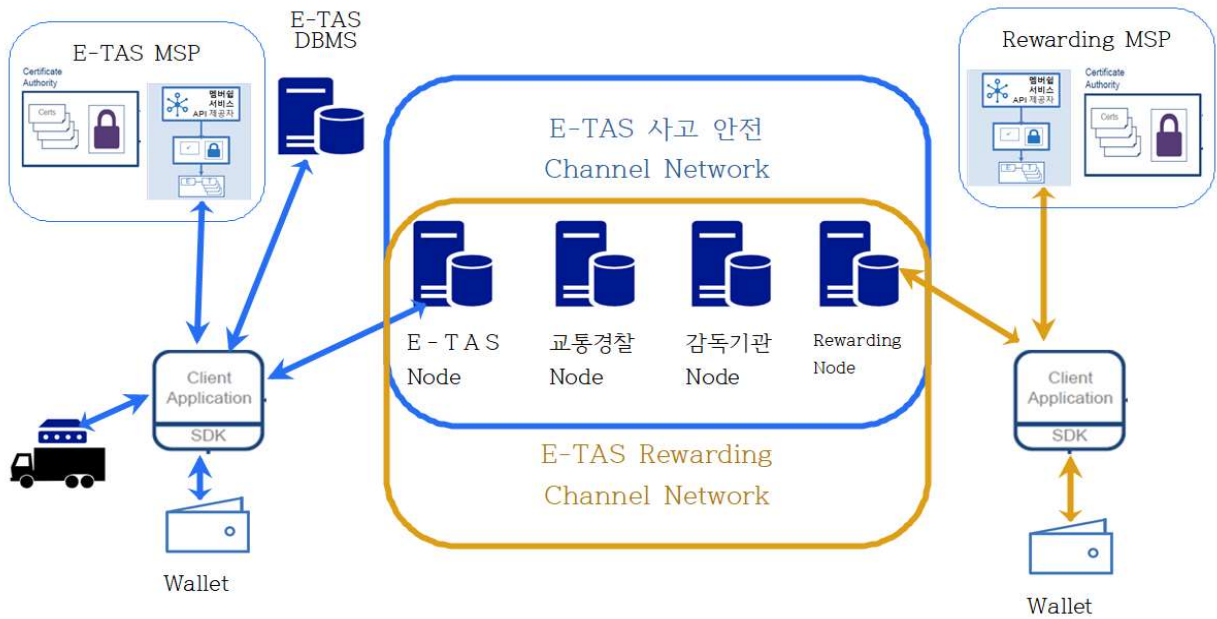
CASE	검증방법	검증기준	검증결과		
목표자동차 정지 상태	● 운행자동차 속도 45km/h로 목표자동차에 30m 간격으로 접근 ● 7회 중 5회 이상 경고	● 전방추돌경고 ● 전방 30m에서 거리표시	No	경고발생	거리
			1	pass	28
			2	pass	27
			3	pass	28
			4	pass	29
			5	fail	32
			6	pass	25
			7	fail	32
목표자동차 감속운행	● 차간거리를 45m로 유지하며, 45km/h 속도로 주행 ● 목표자동차가 감속을 시작하여 차간거리가 약 30m 간격으로 접근하였을 때 경고 ● 7회 중 5회 이상 경고	● 전방추돌경고 ● 전방 30m에서 거리표시	No	경고발생	거리
			1	pass	27
			2	pass	26
			3	fail	31
			4	pass	27
			5	pass	28
			6	pass	29
			7	pass	28
목표자동차 저속운행	● 운행자동차 속도 45km/h로 100m 이상 충분한 간격을 두고 목표자동차를 향해 주행하고, 목표자동차는 30km/h의 일정 속도로 저속주행, 거리가 30m가 되었을 때 경고 ● 7회 중 5회 이상 경고	● 전방추돌경고 ● 전방 30m에서 거리표시	No	경고발생	거리
			1	pass	21
			2	fail	31
			3	pass	29
			4	pass	27
			5	pass	28
			6	pass	29
			7	fail	30

위탁 기관 : 상명대학교

2. 블록체인 보상시스템 <핵심성과물 No.4>

2-1. 사고안전운전 보상 네트워크

사고안전 보상 블록체인 네트워크는 두 개의 네트워크로 구성되어 있다. 첫째 사고안전에 관련된 운행정보와 사고 이력 등을 블록체인 네트워크에 기록 관리하며, 둘째로는 첫 번째의 정보를 바탕으로 안전운전을 유도하는 보상네트워크로 구성되어 있다.



[그림] 사고안전 블록체인 보상네트워크

사고안전 보상 네트워크의 구성을 도표로 나타내면 다음과 같다.

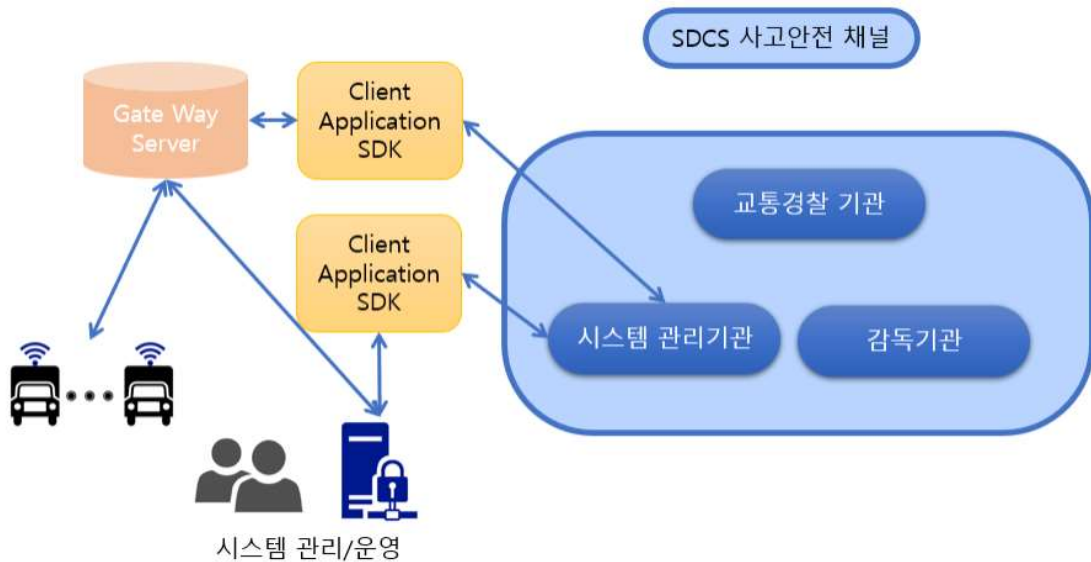
표 10 사고안전 보상네트워크의 구성요소

구분	기능	비고
E-TAS MSP	E-TAS 계정등록 및 권한관리 인증서버	E-TAS MSP는 사고 및 안전운전 정보를 제공하는 네트워크의 운전자 등록정보, 채널가입정보, 개인등록정보를 관리하고 인증하는 기능을 담당한다.
E-TAS DBMS	DTG 디지털 운행기록계 Database	E-TAS 데이터베이스는 DTG에서 업로드된 데이터를 저장하고, 이를 분석하여 도로의 위험운전정보를 분석하여 E-TAS 블록체인 네트워크에 위험운전행위 자료를 기록한다.
Wallet	DTG 단말기에 전자지갑	DTG단말기의 등록정보, 로그인 계정정보, 사용자 등록정보 등 네트워크에 접근할 수 있는 인증서를 함께 포함하고 있다.
E-TAS 사고-안전 채널 네트워크	하이퍼레저 기반의 비공개 블록체인 네트워크.	비공개 블록체인 네트워크로 허가된 네트워크 구성원만 원장을 공유함으로써 도로의 안전정보와 운전자의 위험행동을 원장에 기록하며, 기록된 정보는 감독기관, 단속기관, 리워딩 노드에 운전자, 운수회사 별 누적횟수가 기록된다.
Rewarding MSP	안전운행정보 제공 과 보상 네트워크 사용자의 등록과 권한 관리 인증서버	도로운행을 관리하는 관리자, 운전자, 운수회사 등 사용자 등록정보와 조직정보, 채널가입정보, 개인등록정보, 보상정보 등 인증정보 제공.
E - T A S Rewarding Channel	Rewarding 채널로서 실시간 제공된 안전운전 가이드 정보와 안전운전 실행 내용을 원장에 기록한다.	Rewarding 채널은 안전운전 가이드를 준수하고 위험운전행위를 기록함으로써 교통사고를 예방하고자하는 시스템이다. 따라서 안전운전 결과를 전자지갑에 계정정보와 개인정보와 함께 보상내용을 기록한다.

1) 사고안전운전 보상시스템의 채널별 구성

참여기관	SDCS사고안전채널	SDSC보상채널	비고
운영기관	○	○	
감독기관	○	○	
교통경찰	○		
보상노드		○	

① 사고안전보상네트워크 사고안전 채널의 구성



[그림] 사고안전 네트워크(SDCS)의 구성

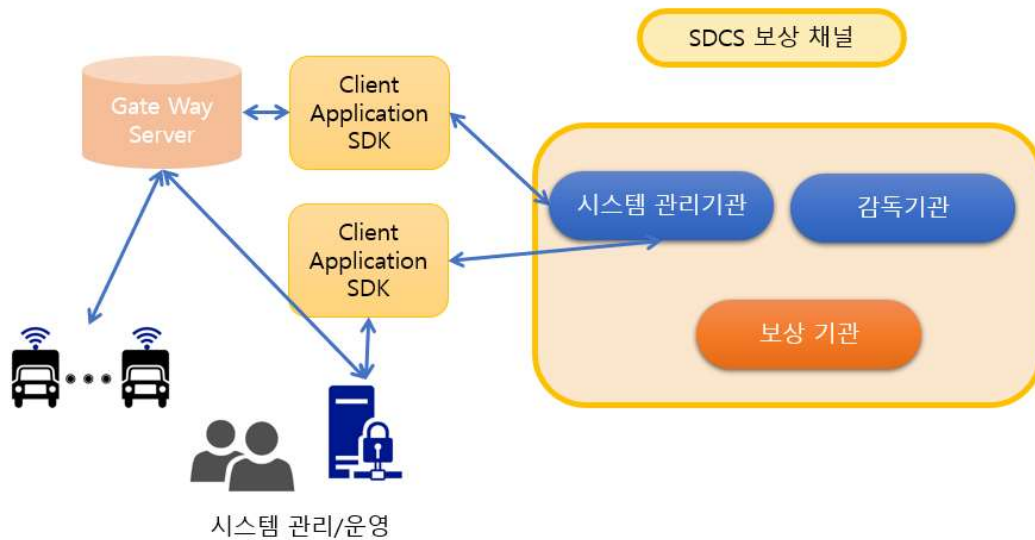
표 11 사고안전보상 네트워크(SDCS)의 사고안전 채널의 구성 내용

구 성	설 명	비고
Gateway Server	블록체인 네트워크를 접속통로제공	
Client SDK	체인코드 설치 및 실행프로그램 개발 도구	
시스템 관리/운영	블록체인 네트워크 오퍼레이터	
시스템관리기관	시스템 관리기관(Ordere)	
감독기관	분산원장 관리기관 감독기관(Org1)	
교통경찰	분산원장 관리기관(Org2)	

사고안전 네트워크는 화물운전자의 안전운전을 유도하기 위한 시스템으로 차량에 설치된 단말기에 사고 안전에 관련된 데이터가 자동으로 수집되어 온/오프라인으로 게이트웨이 서버를 통하여 블록체인 네트워크에 등록된다. 입력되는 데이터는 실시간을 필요로 하지는 않으나, 무선통신 기반의 디지털 운행기록계의 경우 사고안전 데이터 뿐 만 아니라 실시간 위치정보를 제공함으로써 다양한 부가 서비스가 가능해 진다.

2-2. 사고안전운전 보상 네트워크 사고안전 보상네트워크의 보상채널의 구성

사고안전 보상네트워크의 보상채널은 3차연도에 개발예정인 시스템으로 사고안전블록체인 네트워크의 데이터를 바탕으로 보상을 제공하는 구성을 가지고 있다. 사고안전 네트워크와 보상네트워크는 채널로 분리되어 서로의 네트워크에 영향을 주지 않도록 설계되어 있으며, 두 시스템에 공통으로 연결된 관리기관이나 운영기관에서 모니터링 및 제어가 가능하다.



[그림] 사고안전 보상 네트워크의 구성

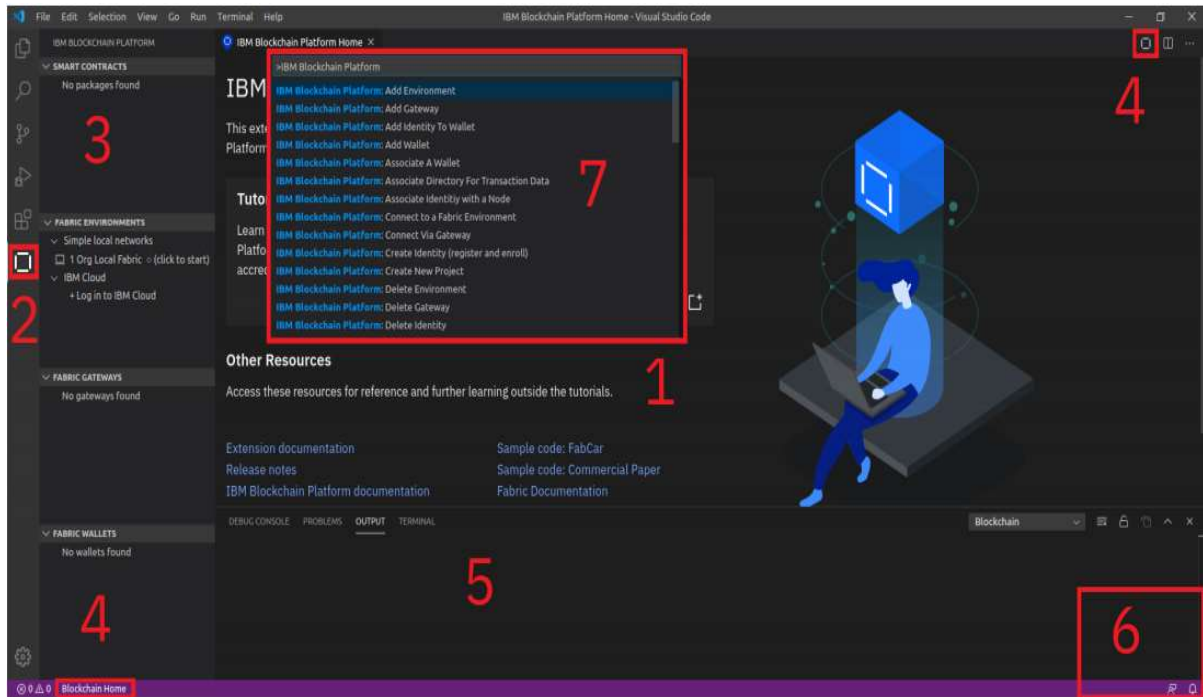
표 12 사고안전 네트워크(SDCS) 보상채널의 구성

구 성	설 명	비고
Gateway Server	블록체인 네트워크를 접속통로제공	
Client SDK	체인코드 설치 및 실행프로그램 개발 도구	
시스템 관리/운영	블록체인 네트워크 오퍼레이터	
시스템 관리기관	시스템 관리기관(Orderer)	
감독기관	분산원장 관리기관 감독기관(Org1)	
보상관련 기관	분산원장 관리기관(Org2)	

2-3. 사고안전 블록체인 네트워크 시스템 개발

① 블록체인 네트워크 개발툴의 선정

사고안전 블록체인 네트워크 시스템 개발은 폐쇄형 블록체인 개발 툴로 잘 알려진 IBM Blockchain Platform을 이용하여 개발 하였으며, IDE 툴은 VisualStudio Code를 이용하였다. 두 시스템 모두 크로스플랫폼 환경을 지원하고 있으며, 블록체인 개발언어 node.js, java, javascript, go, golang 등을 모두 지원 할 수 있는 툴로서 편리함과 신뢰성을 함께 제공하고 있다. 더욱이 IBM Blockchain Platform의 경우는 Stand Alone 환경에서 개발 운영하다가 시스템의 확장시 클라우드 환경으로 이전이 편리하도록 다양한 기능과 편리성을 제공하고 있다.



[그림] VS code에서 IBM Blockchain Platform의 환경설정

VisualStudio Code는 IDE개발지원 툴로서 다양한 프로그램 개발환경을 제공하고 있으며, Microsoft가 개발, 지원 하고 있다. 따라서 지속적인 지원과 버전업이 진행되고 있으며, 확장성과 신뢰성이 높다. 특히 golang, java, javascript, typescript등 Hyperledger Fabric에 사용되는 블록체인개발 언어를 모두 사용할 수 있어 편리한 도구가 된다.

표 13 VS code IDE 프로그램 화면 구성

구분	내용	비고
1.Editor pane	Smart Contract 및 어플리케이션 코드 편집화면	
2.IBM ICON	IBM Blockchain Platform 의 활성화버튼	
3.Blockchain side bar	Smart Contract 및 체인코드 패키징 화면	
4.Fabric Wallet	사용자 계정 및 인증서가 포함된 전자지갑	
5.Output	출력화면, 터미널 화면, 디버깅 화면 등 다양한 출력환경 지원	
6.Notification	알림화면으로 프로그램 수행시 발생하는 오류등을 표시함	
7.Command Palette	확장프로그램, 추가메뉴 등 다양한 환경제어를 수행.	

- Editor pane : source code 편집을 수행하는 화면으로 주로 사용되며, 플러그인 소프트웨어의 설치, 확장프로그램 설명을 제공한다.
- IBM Blockchain Platform 실행 아이콘 : Visualstudio Code를 블록체인 개발환경으로 전환하는 실행아이콘으로 블록체인 네트워크의 구성, 체인코드의 개발, 설치, 실행 등의 모든 개발환경을 제공한다.
- Blockchain side bar: 블록체인 네트워크에서 사용되는 스마트계약인 체인코드 목록을 나타낸다. 이곳에 표시되는 체인코드는 블록체인 피어에 설치되기 전의 상태인 패키징 상태의 프로그램이다.
- Fabric Wallet : 블록체인 네트워크에 접속할 자격이 있는 사용자를 관리하는 전자 지갑으로 인증서와 계정정보를 포함한다.
- Output: IBM Blockchain Platform 홈페이지를 연결하는 정보와 아이콘으로 개발 툴의 특징과 사용방법 등을 설명한 정보제공창
- Notification : 블록체인네트워크 운영과정에서 발생하는 알림을 표시하는 창이며, 알림의 기록이 보관된다.
- Command palette.: 블록체인을 운용하는 docker, docker-compose, npm, nodejs와 같은 툴의 제어하는 명령어를 실행하는 명령실행 창.

② 개발환경에 필요한 요구조건

1) 운영체제 linux (ubuntu 18.04LST)

-node.js (>=15.3 or 12.15.0)

-npm (>= 6.0.0)

-docker (>=17.6.2)

-ocker-composer (>=1.14.0)

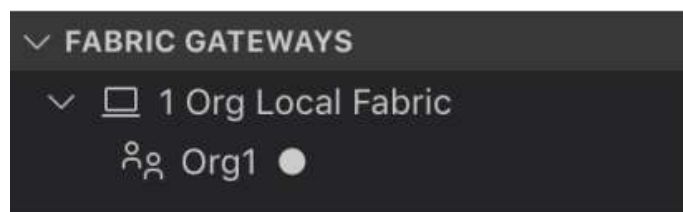
-go 1.14.7

블록체인 네트워크의 개발환경은 리눅스 배포판중 많이 사용되고 있는 ubuntu 18.04LST에서 개발하였다. 하이퍼레저패브릭의 개발은 golang으로 개발되었으며, Client SDK 의 경우는 golang, java, nodejs 등 다양한 언어로 개발되고 있다. 또한 셸스크립트 환경에서 블록체인 네트워크를 운용하기 위한 Client CLI SDK 도 지원되고 있어 개발자의 환경에 적합한 툴을 선택할 수 있도록 지원하고 있다. 블록체인 네트워크의 구성이 여러 대의 서버들이 복합적으로 운영되어야 하며, 이를 관리하는 것이 어려운 문제이다. 이를 관리하기 위한 툴로서 docker와 coker-composer 가 사용되고 있으며, 위에 나열한 프로그램의 버전은 IBM Blockchain Platform 환경에 필요한 지원 소프트웨어의 버전을 나타내고 있다. 버전이 맞지 않으면 실행 중 의도하지 않은 오류를 발생시키며, 프로그램 설치가 진행되지 않는 경우가 있어 주의를 기울여야 한다.

③ 블록체인 네트워크 접속도구 (Connection Profile)

블록체인 네트워크에 접속하기 위해서는 계정 (Identity), 신원증명 파일이 포함된 전자지갑(Wallet), 게이트웨이(Gateway) 접속 프로파일 정보가 있어야 한다. 이러한 신원증명에 관련된 증명서의 발행은, 네트워크에 소속된 기관에서(Certificate Authority) 과 MSP에서 계정정보와 인증서를 발행한다. 더욱이 네트워크에 포함된 모든 구성 요소들은 각자의 신원정보(Identity)를 가지고 있으며, 네트워크에서 발행되는 지갑(Wallet)에 신원정보를 포함할 수 있다.

블록체인 네트워크에 접속하기 위해서는 아래의 화면에서와 같이 신원증명을 통해 게이트웨이로 접속을 하여야 한다. 즉 계정정보(Identity),신원증명을 위한 (Wallet)을 정보를 네트워크 접속 프로파일에 포함하여 게이트웨이에 접속한다.



[그림] Hyperledger Fabric Gate Way



[그림] 전자지갑에 포함된 신원정보(Identity)

네트워크에서는 하나의 지갑에 포함되는 신원정보를 2가지를 가지고 있다.

Orderer Wallet
-admin: Orderer node에 관련된 신원정보
-ordererAdmin: Ordering Service에 사용되는 신원정보.
Org1 wallet
사용자가 가장 많이 사용하는 전자지갑으로
admin: Org1Peer 노드에 관련된 신원정보
org1Admin: 이 신원정보는 submit트랜잭션이나, evaluate Transaction을 수행할 때 사용되는 신원정보이다.

따라서 게이트웨이 접속을 할 경우 org1Admin 계정을 이용하여 게이트웨이에 접속한다. 좌측 화면에서 표시된 정보로부터 게이트웨이(Gateway)에 접속에서 나타나는 정보는 Identity org1Admin의 개인증명으로 스마트계약 demo-contract@0.0.1 의 버전정보, 접속신원, 실행 가능한 체인코드 리스트가 함께 제공된다.

④ 기본템플릿으로 제공되는 체인코드

표 14 기본 템플릿으로 제공되는 체인코드 설명

체인코드명칭	명령타입	입력형식	비고
myAssetExite	evaluate type	[“001”]	킷값의 해당하는 자료가 있는지 확인
createMyAsset	submit type	[“001”, “Mona Lisa”]	001키에 “Mona Lisa” 입력
readMyAsset	evaluatin type	[“001”]	“Mona Lisa” 킷값 반환
updateMyAsset	submit type	[“001”, “Da Vinci”]	“Mona Lisa” -> “Da Vinci”로 변경
deleteMyAsset	submit type	[“001”]	[“001”]키값을 갖는 월드상태 삭제

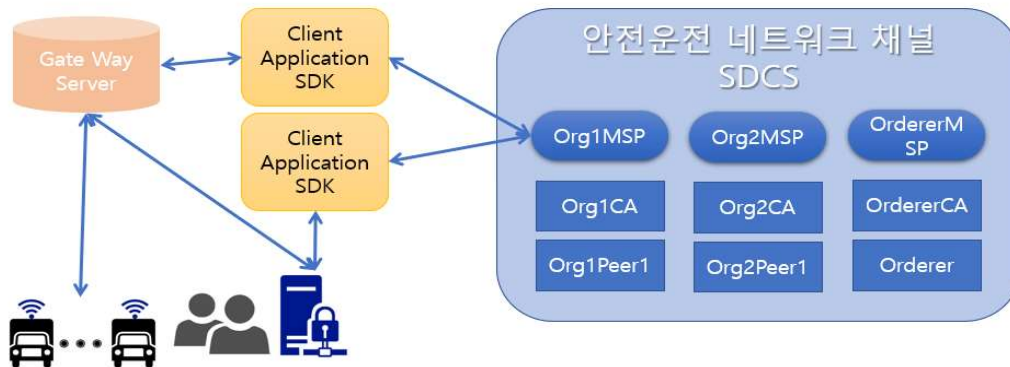
기본템플릿으로 설치된 체인코드(Smart Contract)는 5가지의 함수가 제공된다. 함수 이름에서 알 수 있듯이 블록체인 네트워크에 월드상태(World state)상태의 데이터를 검색, 변경, 삭제 할 수 있는 명령어 집합이다. myAssetExite 매개변수로서 key : string 값의 형태로 [“001”] 001키 값에 해당하는 데이터의 정보를 출력하라는 명령이다.

createMyAsset 체인코드는 새로운 데이터를 생성하는 데이터 처리로서 월드상태 정보가 추가되는 처리 루틴을 포함하고 있어 submit 타입으로 분류한다. 입력데이터로는 키 값과 추가되는 정보를 모두 포함하는 입력데이터를 작성하여야 한다. readMyAsset 체인코드는 킷값을 갖는 데이터의 정보를 반환하는 코드이다. updateMyAsset체인코드는 현재 네트워크에 저장되어 있는 데이터를 변경할 목적으로 사용된다. 따

라서 킷값과 변경할 내용 모두를 포함하고 있어야 프로그램을 수행할 수 있으며, createMyAsset 체인코드와 기능이 거의 같은 명령이다. 마지막으로 deleteMyAsset 체인코드는 주어진 킷값에 해당하는 정보를 삭제하도록 하는 명령이다.

⑤ 안전운전 보상네트워크의 사고안전 채널 구성

안전운전 보상네트워크의 구성은 3개의 조직과 1개의 채널로 구성되어 있는 블록체인 네트워크이다. 네트워크의 구성을 살펴보면 각각의 조직마다. MSP 서비스와 CA 서비스를 제공하고 있어 그룹에 해당하는 사용자들에게 편리한 서비스를 제공하고 있다.



[그림] 안전운전 블록체인 네트워크(SDCS)

표 15 안전운전보상 네트워크(SDCS)의 사고안전 채널의 구성 내용

구 성	설 명	비고
Gateway Server	블록체인 네트워크를 접속통로제공	
Client SDK	체인코드 설치 및 실행프로그램 개발 도구	
시스템 관리/운영	블록체인 네트워크 오퍼레이터	
Orderer	시스템 관리기관(Orderer)	
Org1	분산원장 관리기관 감독기관(Org1)	
	Org1MSP (조직의 역할 관리)	
	Org1CA (Org1의 인증서 발행)	
	Org1Peer1 (분산원장관리 노드)	
Org2	분산원장 관리기관 감독기관(Org2)	
	Org2MSP (조직의 역할 관리)	
	Org2CA (Org2의 인증서 발행)	

표 16 안전운전 변수명칭 및 데이터

순서	변수명칭	값	비고
1	no	0001	64bit 연번
2	trip_key	C118436389719042008020600	
3	dtg_model	QDTG-S+	
4	car_vin	WVFGL23C0EM123819	
5	car_nq	전북80사2150	
6	trans_reg_no	7481234521	
7	drvier_code	0000000000	
8	tachmeter_day	0	
9	tachmeter	39953	
10	speed	30	
11	rpm	600	
12	break_on	0	
13	gps _ x	127.040368	
14	gps _ y	37.462854	
15	azimuth	290	
16	vx	-0.65	
17	vy	0.30	
18	status	00	
19	region_code	47	
20	company_location_code	17	
21	company_code	00331	
22	date	20091410050011	
23	fcws	0	
24	ldws	0	
25	sdns	0	
26	sldws	0	

2-4. 블록체인 네트워크 체인코드 설계

① 자산(Asset)으로 데이터셋 구성

안전운전에 관련된 데이터를 처리하는데 있어 총 26개의 속성 값을 가지고 있다. 이들 속성값은 체인코드를 실행할 때 기본 입출력 데이터구조이며, 블록체인 데이터를 활용하는 다양한 체인코드에서의 활용의 관점에서 구성하여야 한다.

표 17 자산데이터 셋(MyAsset class)

```
import { Object, Property } from 'fabric-contract-api';
@Object()
export class MyAsset {
  @Property()
  public id: string;
  @Property()
  public date: string;
  @Property()
  public trip_key: string;
  @Property()
  public dtg_model: string;
  @Property()
  public car_vin: string;
  @Property()
  public car_nq: string;
  @Property()
  public trans_reg_no: string;
  @Property()
  public driver_code: string;
  @Property()
  public tachmeter_day: string;
  @Property()
  public tachmeter: string;
```

```

@property()
public speed: string;
@property()
public rpm: string;
@property()
public break_on: string;
@property()
public gps_x: string;
@property()
public gps_y: string;
@property()
public azimuth: string;
@property()
public vx: string;
@property()
public vy: string;
@property()
public status: string;
@property()
public region_code: string;
@property()
public company_location: string;
@property()
public company_code: string;
@property()
public tcws: string;
@property()
public ldws: string;
@property()
public sdns: string;
@property()
public sldws: string;
@property()
public region_code: string;
@property()
public company_location: string;
@property()
public company_code: string;
@property()
public tcws: string;
@property()
public ldws: string;
@property()
public sdns: string;
@property()
public sldws: string;
}

```

② myAssetExist

```

@Transaction(false)
@Returns('boolean')
public async myAssetExists(ctx: Context, myAssetId: string): Promise<boolean> {
    const buffer = await ctx.stub.getState(myAssetId);
    return (!!buffer && buffer.length > 0);
}

```

③ CreateMyAsset

```

@Transaction()
public async createMyAsset(ctx: Context, id: string, value: string, id: string,
    date: string, trip_key: string, dtg_model: string, car_vin: string, car_nq: string,
    trans_reg_no: string, driver_code: string, tachmeter_day: string, tachmeter: string,
    speed: string, rpm: string, break_on: string, gps_x: string, gps_y: string, azimuth:
    string, vx: string, vy: string, status: string, region_code: string,
    company_location: string, company_code: string, tcws: string, ldws: string, sdns:
    string, sldws: string ): Promise<void> {
    ): Promise<void> {
        const exists = await this.myAssetExists(ctx, id);
        if (exists) {

```

```

        throw new Error(`The my asset ${id} already exists`);
    }
    const asset = {
      MyAssetId:myAssetId ,
      Date:date ,
      Trip_kye:trip_key ,
      Dtg_model:dtg_model ,
      Car_vin:car_vin ,
      Car_nq:car_nq ,
      Trans_reg_no: trans_reg_no ,
      Driver_code:driver_code ,
      tachmeter_day: tachmeter_day ,
      Tachmeter:tachmeter ,
      Speed: speed ,
      Rpm:rpm ,
      Break_on:break_on ,
      Gps_x:gps_x ,
      Gps_y:gps_y: ,
      Azimuth:azimuth ,
      Vx:vx ,
      Vy:vy ,
      Status:status,
      Region_code:region_code: ,
      Company_location: company_location ,
      Company_code: company_code ,
      Tcws: tcws ,
      Ldws:ldws ,
      Sdns:sdns ,
      Sldws:sldws
    };
    const buffer = Buffer.from(JSON.stringify(asset));
    await ctx.stub.putState(id, buffer);
  }

```

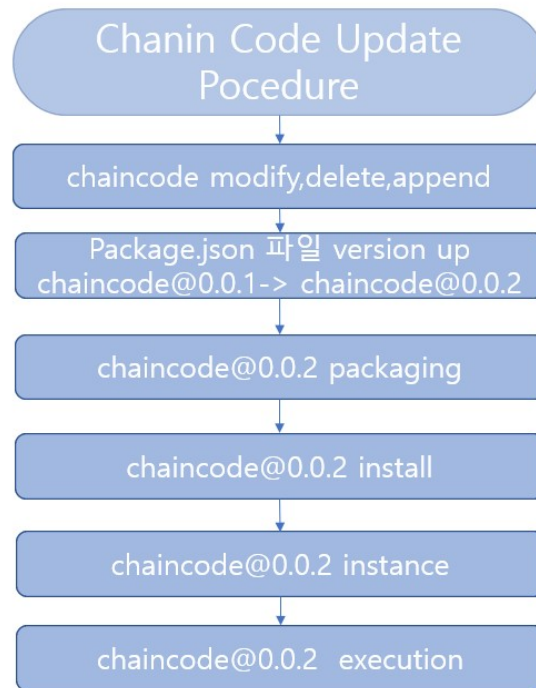
④ UpdateMyAsset

```

@Transaction()
public async updateMyAsset(ctx: Context, myAssetId: string,
date: string, trip_key: string, dtg_model: string, car_vin: string, car_nq: string,
trans_reg_no: string, driver_code: string, tachmeter_day: string, tachmeter: string,
speed: string, rpm: string, break_on: string, gps_x: string, gps_y: string, azimuth: string,
vx: string, vy: string, status: string, region_code: string, company_location: string,
company_code: string, tcws: string, ldws: string, sdns: string, sldws: string ):Promise<void>
{
    const exists = await this.myAssetExists(ctx, myAssetId);
    if (exists) {
        throw new Error(`The my asset ${myAssetId} already exists`);
    }
    const asset = {
        MyAssetId: myAssetId ,
        Date: date ,
        Trip_key:trip_key ,
        Dtg_model:dtg_model ,
        Car_vin:car_vin ,
        Car_nq:car_nq ,
        Trans_reg_no: trans_reg_no ,
        Driver_code:driver_code ,
        Tachmeter_day: tachmeter_day ,
        Tachmeter:tachmeter ,
        Speed: speed ,
        Rpm:rpm ,
        Break_on:break_on ,
        Gps_x:gps_x ,
        Gps_y:gps_y ,
        Azimuth:azimuth ,
        Vx:vx , Vy:vy ,
        Status:status,
        Region_code:region_code ,
        Company_location: company_location ,
        Company_code: company_code ,
        Tcws: tcws ,
        Ldws:ldws ,
        Sdns:sdns ,
        Sldws:sldws,
    };
    const buffer = Buffer.from(JSON.stringify(asset));
    await ctx.stub.putState(myAssetId, buffer);
}

```

2-5. 체인코드 업데이트 절차



[그림] 체인코드 업데이트 절차

- ① 프로그램 업그레이드
- ① package.json 파일 버전업그레이드
- ② 체인코드 패키징
- ③ 패키징된 체인코드 피어에 설치
- ④ 설치된 체인코드 인스턴스
- ⑤ 체인코드 실행

2-6. 체인코드 실행결과

① myAssetExists

항목	내용	
체인코드 이름	myAssetTxists	
체인코드 형식	evaluation transaction type	
체인코드 기능	블록체인 네트워크에 해당 키값의 월드상태가 있는지 확인하는 체인코드	

입력 내용: 001 key 값을 갖는 월드상태 데이터 호출 (World state)

```

["001"]
optional: What are the arguments to the transaction, (e.g. ["arg1", "arg2"]) (Press 'Enter' to
confirm or 'Escape' to cancel)
  
```

실행결과

evaluate Transaction myAssetExists 실행결과 mychannel에는 ["001"] 트랜잭션 결과 값이 존재함을 나타낸다.

```

[INFO] evaluateTransaction
[INFO] evaluating transaction myAssetExists with args 001 on channel mychannel
[SUCCESS] Returned value from myAssetExists: true
  
```

② createMyAsset

항목	내용	
체인코드 이름	createMyAsset	
체인코드 형식	submit transaction type	
체인코드 기능	블록체인 네트워크에 새로운 데이터를 등록하는 트랜잭션 체인코드	

입력내용:

```
["011","20091410050011","C118436389719042008020600","QDTG-S+","WVFG23C0EM123819","
optional: What are the arguments to the transaction, (e.g. ["arg1", "arg2"]) (Press 'Enter' to
confirm or 'Escape' to cancel)
```

```
["011","20091410050011","C118436389719042008020600","QDTG-S+","WVFG23C0EM123
819","전북80사2150","7481234521","0","0","39953","30","600","0","127.040368","37.462854
","290","1.74","-0.86","0","47","17","331","0","0","0","0"]
```

실행결과

["011"]키값을 갖는 데이터를 createMyAsset 체인코드를 이용하여 기록한 결과 mychannel 에 기록됨을 나타낸다.

```
[INFO] submitTransaction
[INFO] submitting transaction createMyAsset with args 011,20091410050011,C118436389719042008020600,QDTG-S+,WVFG23C0EM123819,전북80사2150,
,127.040368,37.462854,290,1.74,-0.86,0,47,17,331,0,0,0,0 on channel mychannel
[SUCCESS] No value returned from createMyAsset
[INFO] submitTransaction
```

③ readMyAsset

항목	내용	
체인코드 이름	readMyAsset	
체인코드 형식	evaluation transaction type	
체인코드 기능	블록체인 네트워크에 키값의 월드상태의 데이터를 반환하는 체인코드	

입력데이터

["010"] 010 키값을 갖는 월드상태 값을 반환하는 체인코드

```
["010"]
optional: What are the arguments to the transaction, (e.g. ["arg1", "arg2"]) (Press 'Enter' to
confirm or 'Escape' to cancel)
```

실행결과 readMyAsset 체인코드 실행 결과

```
[INFO] evaluateTransaction
[INFO] evaluating transaction readMyAsset with args 010 on channel mychannel
[SUCCESS] Returned value from readMyAsset: {"Azimuth":"290","Break_on":"0","Car_nq":"전북80사2150",
```

["010"] 키값을 갖는 월드상태 데이터

```
{ "Azimuth": "290", "Break_on": "0", "Car_nq": "전북80사2150", "Car_vin": "WVFG23C0EM123819", "Company_code": "331", "Company_location": "17", "Date": "20091410050011", "Driver_code": "0", "Dtg_model": "QDTG-S+", "Gps_x": "127.040368", "Gps_y": "37.462854", "Ldws": "0", "Region_code": "47", "Rpm": "600", "Sdns": "0", "Sldws": "0", "Speed": "30", "Status": "0", "Tachmeter": "39953", "Tachmeter_day": "0", "Tcws": "0", "Trans_reg_no": "7481234521", "Trip_key": "C118436389719042008020600", "Vx": "1.74", "Vy": "-0.86" }
```

④ queryRangeAssets

항목	내용	
체인코드 이름	queryRangeAssets	
체인코드 형식	evaluation transaction type	
체인코드 기능	시작키값-종료키값의 영역에 해당하는 월드상태 데이터를 반환하는 트랜잭션 체인코드	

입력내용

[“001”, “010”] 001에서 010 키값 까지의 월드상태(World state) 값을 반환하는 체인코드

```
[ "001", "010" ]
optional: What are the arguments to the transaction, (e.g. ["arg1", "arg2"]) (Press 'Enter' to confirm or 'Escape' to cancel)
```

실행결과

[“001”, “010”] 001 키값에서 010키값 사이의 데이터 출력 결과

```
[INFO] evaluateTransaction
[INFO] evaluating transaction queryRangeAssets with args 001,010 on channel mychannel
[SUCCESS] Returned value from queryRangeAssets: [{"Key": "001", "Record": {"Azimuth": "290", "Break_on": "0", "Car_nq": "전북80사2150",
```

```
19042008020600","Vx":"1.74","Vy":"-0.86"}},{ "Key":"003","Record":{"Azimuth":"290","Break_on":"0","Car_nq":"전북80사2150","Car_vin":"WVFG23C0EM123819","Company_code":"331","Company_location":"17","Date":"20091410050011","Driver_code":"0","Dtg_model":"QDTG-S+","Gps_x":"127.040368","Gps_y":"37.462854","Ldws":"0","Region_code":"47","Rpm":"600","Sdns":"0","Sldws":"0","Speed":"30","Status":"0","Tachmeter":"39953","Tachmeter_day":"0","Tcws":"0","Trans_reg_no":"7481234521","Trip_key":"C118436389719042008020600","Vx":"1.74","Vy":"-0.86"}},{ "Key":"004","Record":{"Azimuth":"290","Break_on":"0","Car_nq":"전북80사2150","Car_vin":"WVFG23C0EM123819","Company_code":"331","Company_location":"17","Date":"20091410050011","Driver_code":"0","Dtg_model":"QDTG-S+","Gps_x":"127.040368","Gps_y":"37.462854","Ldws":"0","Region_code":"47","Rpm":"600","Sdns":"0","Sldws":"0","Speed":"30","Status":"0","Tachmeter":"39953","Tachmeter_day":"0","Tcws":"0","Trans_reg_no":"7481234521","Trip_key":"C118436389719042008020600","Vx":"1.74","Vy":"-0.86"}},{ "Key":"006","Record":{"Azimuth":"290","Break_on":"0","Car_nq":"전북80사2150","Car_vin":"WVFG23C0EM123819","Date":"20091410050011","Driver_code":"0","Dtg_model":"QDTG-S+","Trans_reg_no":"7481234521","Trip_key":"C118436389719042008020600"}},{ "Key":"007","Record":{"Azimuth":"290","Break_on":"0","Car_nq":"전북80사2150","Car_vin":"WVFG23C0EM123819","Company_code":"331","Company_location":"17","Date":"20091410050011","Driver_code":"0","Dtg_model":"QDTG-S+","Gps_x":"127.040368","Gps_y":"37.462854","Ldws":"0","Region_code":"47","Rpm":"600","Sdns":"0","Sldws":"0","Speed":"30","Status":"0","Tachmeter":"39953","Tachmeter_day":"0","Tcws":"0","Trans_reg_no":"7481234521","Trip_key":"C118436389719042008020600","Vx":"1.74","Vy":"-0.86"}},{ "Key":"008","Record":{"Azimuth":"290","Break_on":"0","Car_nq":"전북80사2150","Car_vin":"WVFG23C0EM123819","Company_code":"331","Company_location":"17","Date":"20091410050011","Driver_code":"0","Dtg_model":"QDTG-S+","Gps_x":"127.040368","Gps_y":"37.462854","Ldws":"0","Region_code":"47","Rpm":"600","Sdns":"0","Sldws":"0","Speed":"30","Status":"0","Tachmeter":"39953","Tachmeter_day":"0","Tcws":"0","Trans_reg_no":"7481234521","Trip_key":"C118436389719042008020600","Vx":"1.74","Vy":"-0.86"}},{ "Key":"009","Record":{"Azimuth":"290","Break_on":"0","Car_nq":"전북80사2150","Car_vin":"WVFG23C0EM123819","Company_code":"331","Company_location":"17","Date":"20091410050011","Driver_code":"0","Dtg_model":"QDTG-S+","Gps_x":"127.040368","Gps_y":"37.462854","Ldws":"0","Region_code":"47","Rpm":"600","Sdns":"0","Sldws":"0","Speed":"30","Status":"0","Tachmeter":"39953","Tachmeter_day":"0","Tcws":"0","Trans_reg_no":"7481234521","Trip_key":"C118436389719042008020600","Vx":"1.74","Vy":"-0.86"}]}
```

⑤ updateMyAsset

항목	내용	
체인코드 이름	updateMyAsset	
체인코드 형식	submit transaction type	
체인코드 기능	주어진 킷값의 월드상태 데이터를 업데이트 하도록 하는 체인코드	

["010"] 키 값을 을 갖는 블록 데이터값을 수정할 때 사용하는 체인코드

입력데이터

```
["010","20091410050011","C118436389719042008020600","QDTG-S+","WVFG23C0EM123819","전북80사2150","7481234521","0","0","39953","30","600","0","127.040368","37.462854","290","1.74","-0.86","0","47","17","331","1","0","0","0"]
```

updateMyAsset 체인코드 입력화면

```
[INFO] evaluateTransaction
[INFO] evaluating transaction readMyAsset with args 010 on channel mychannel
[SUCCESS] Returned value from readMyAsset: {"Azimuth":"290","Break_on":"0","Car_nq":"전북80사2150",
```

실행결과 010키값을 갖는 월드상태가 변경되었음을 나타냄

```
["010","20091410050011","C118436389719042008020600","QDTG-S+","WVFGL23C0EM123819","  
optional: What are the arguments to the transaction, (e.g. ["arg1", "arg2"]) (Press 'Enter' to  
confirm or 'Escape' to cancel)
```

updateMyAsset 체인코드 실행결과 값

```
{"Azimuth":"290","Break_on":"0","Car_nq":"전북80사2150","Car_vin":"WVFGL23C0  
EM123819","Company_code":"331","Company_location":"17","Date":"20091410050  
011","Driver_code":"0","Dtg_model":"QDTG-S+","Gps_x":"127.040368","Gps_y":"3  
7.462854","Ldws":"0","Region_code":"47","Rpm":"600","Sdns":"0","Sldws":"0","Spee  
d":"30","Status":"0","Tachmeter":"39953","Tachmeter_day":"0","Tcws":"0","Trans_r  
eg_no":"7481234521","Trip_key":"C118436389719042008020600","Vx":"1.74","Vy  
":"-0.86"}  
[2020. 10. 22. 오후 11:27:43] [INFO] submitTransaction
```

⑥ deleteMyAsset

항목	내용	
체인코드 이름	deleteMyAsset	
체인코드 형식	submit transaction type	
체인코드 기능	주어진 키값의 월드상태 데이터 셋을 삭제하도록 하는 체인코드 월드상태가 삭제되는것이지 월드상태를 기록했던 트랜잭션 이 삭제되는 것은 아니다.	

입력값

["010"]키값을 갖는 월드상태를 삭제하는 체인코드

```
["010"]  
optional: What are the arguments to the transaction, (e.g. ["arg1", "arg2"]) (Press 'Enter' to  
confirm or 'Escape' to cancel)
```

deleteMyAsset 입력화면

실행결과

["010"] 키값을 갖는 월드상태가 삭제되었음을 나타내는 결과

```
[INFO] submitTransaction  
[INFO] submitting transaction deleteMyAsset with args 010 on channel mychannel  
[SUCCESS] No value returned from deleteMyAsset
```

3차년도 (2021) 연구개발

디지털운행기록단말기는 5인치 터치 디스플레이, 메인프로세서, 가속도센서, GPS센서, 스피커, 카메라, 저장장치, LTE 모뎀 등으로 구성되어있다. 가속도는 9축 센서 데이터 중 가속도 3축 데이터, GPS는 Application Layer의 GPRMC Sentence를 사용하며 위도, 경도, 방위각 데이터를 추출한다. FHD카메라를 통해 이미지를 전송하고 스피커는 ADAS기능에서 이벤트가 발생할 경우 경고음을 출력한다.

단말기는 차량운행기록 프로그램과 영상처리 프로그램 2개가 실행된다. 차량운행기록 프로그램은 OBD동글, 가속도, GPS 센서로부터 데이터를 처리 후 1초마다 해당날짜.TXT 제목으로 데이터를 저장한다. 영상처리 프로그램은 카메라에서 전송된 영상을 추론하여 전방차량 추돌방지, 차선이탈 경보, 규정차선 확인, 중앙선 침범, 급차로 변경, 불법유턴 기능을 판별한다. 동시에 DTG 프로그램에서 전처리 된 데이터를 GRPC 프로토콜을 통해 전송하며, 영상처리 프로그램은 데이터를 활용하여 동작한다.

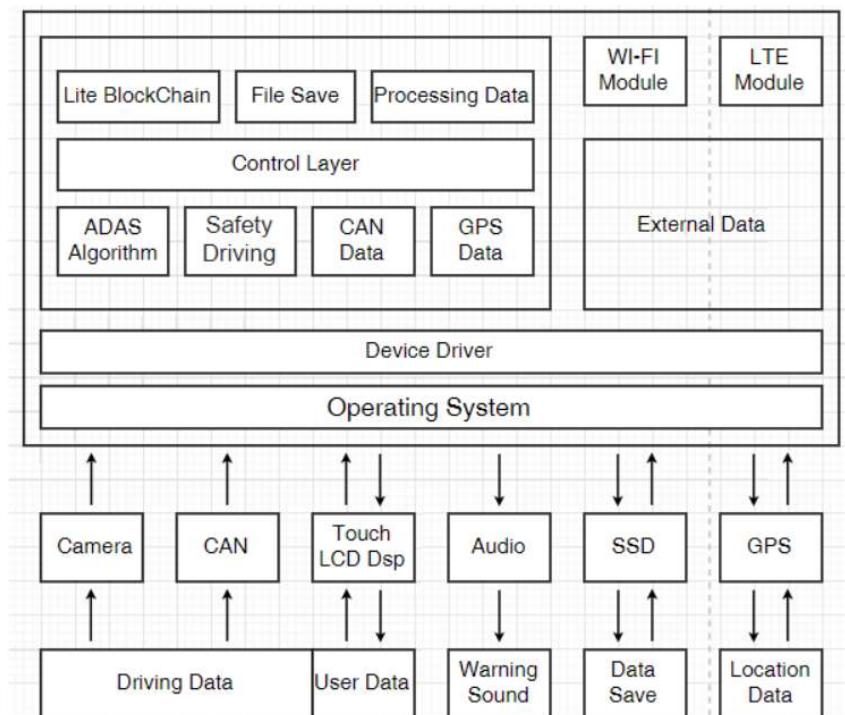


그림 65 디지털운행기록단말기 구성

LTE모뎀을 통해서 디지털운행기록단말기의 데이터가 전송되며, 차량관제서버에서 수신한다. 차량관제서버는 수신된 데이터를 DBMS의 연동하며, WEB과 모바일에서 데이터 요청이 들어오면 웹 서버를 통하여 응답해주며, 데이터를 표출해준다. 또한 안전운행 및 보상정보 블록체인 네트워크와 통신하여 데이터 위·변조를 방지한다.

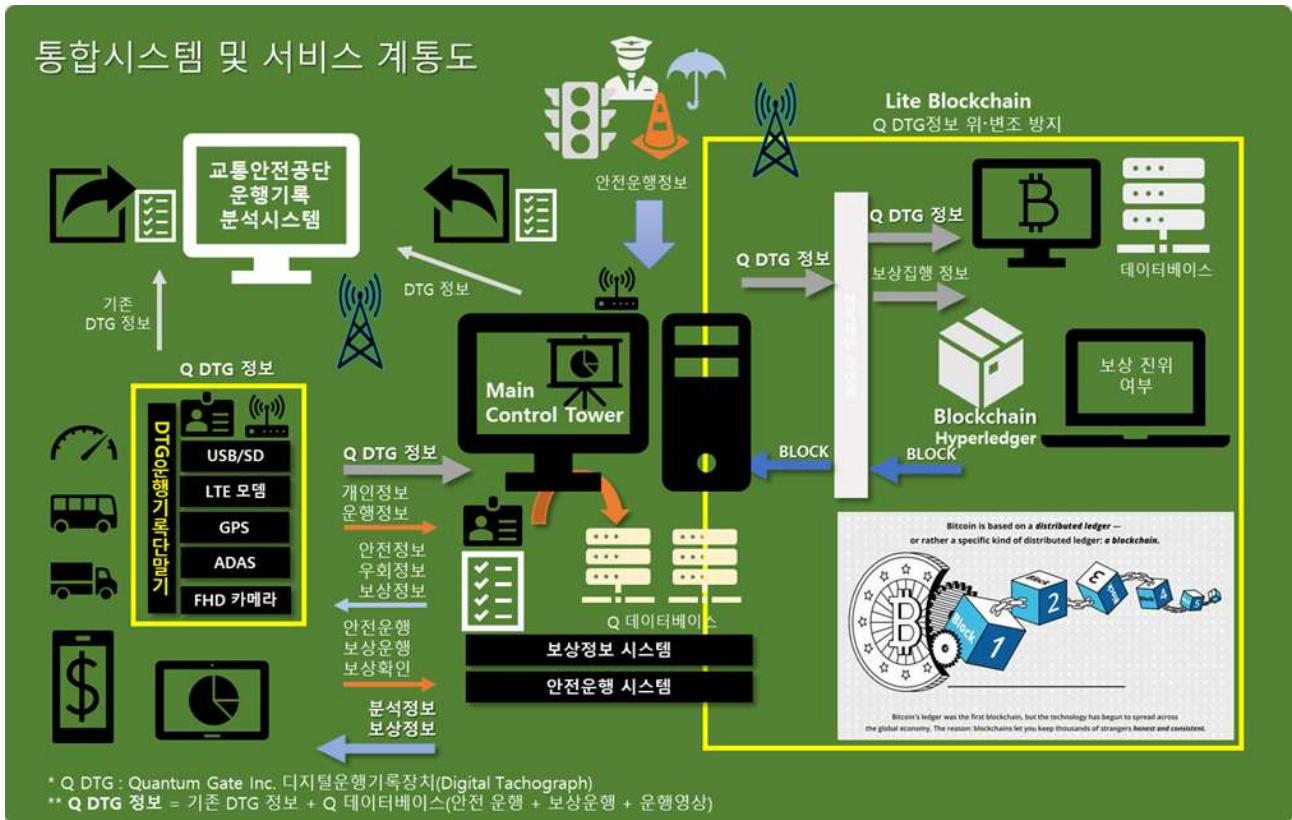


그림 66 DTG 통합시스템 및 서비스 구성도

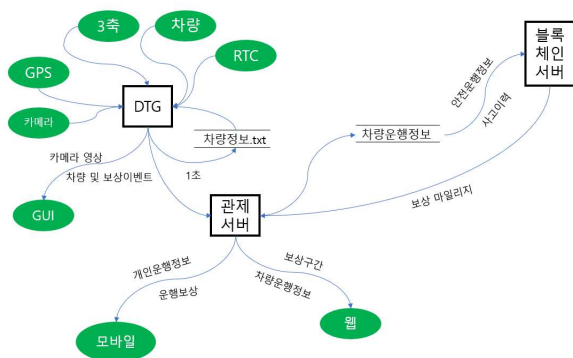


그림 67 데이터 흐름도

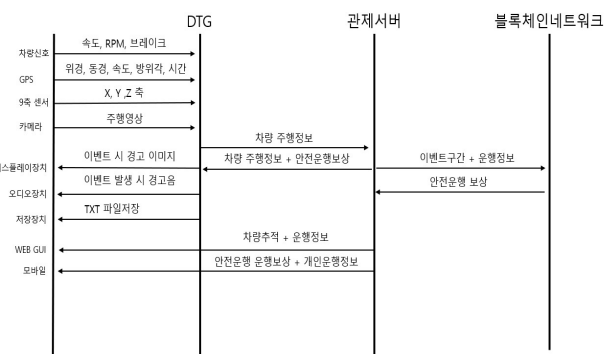
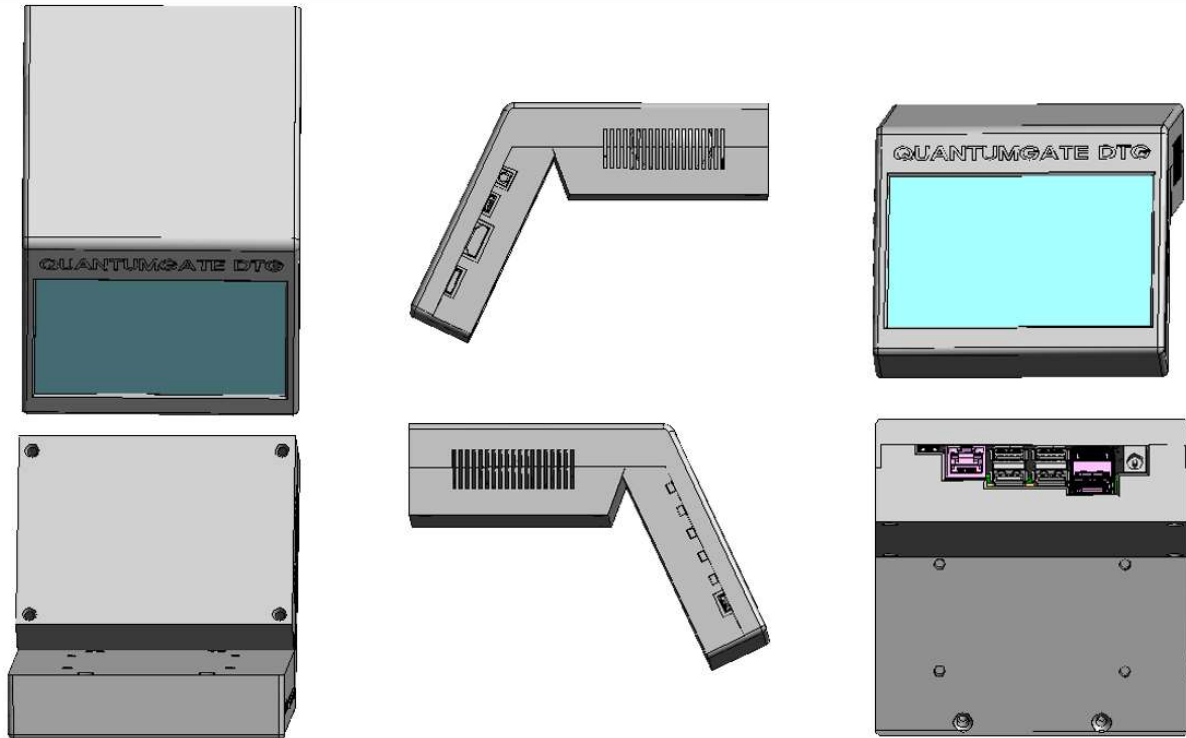


그림 68 시퀀스 흐름도



[표 4] 디지털운행기록단말기

Specification	Picture
• Operating System : Ubuntu 18.04.6 LTS bionic • Kernul : Linux jetson-nano 4.9.201-tegra • CPU : 64-bit Quad-core ARM A57 (1.43GHz) • Memory : 8GB • GPU : 128-core NVIDIA Maxwell • MicroSD SD : 128GB	

[표 5] 차량관제서버

Specification	Picture
• Operating System : Ubuntu 18.04.6 LTS bionic • Kernul : Linux version 5.4.0-90-generic • CPU : Intel(R) Core(TM) i5-6400 @2.70GHz • Memory : 16GB • VGA : NVIDIA GT218(GeForce 210) • SSD : 256GB	

[표 6] 블록체인 네트워크 서버

Specification	Picture
---------------	---------

- Operating System : Ubuntu 18.04.6 LTS bionic
- Kernel : Linux version 5.4.0-90-generic
- Intel(R) Core(TM) i9-9980XE CPU @ 3.00GHz
- Memory : 64GB
- VGA : NVIDIA GP104(GeForce GTX 1080 8GBs)
- SSD : 1TB
- HDD : 2TB



시스템 S/W개발

[차량관제 서버개발]

1. 차량운행데이터 저장용 DB생성

차량관제서버에서 수신된 데이터를 저장할 수 있는 데이터베이스 및 테이블이 생성되는지 확인한다.

```
mysql> show tables;
+-----+
| Tables_in_testdb |
+-----+
| compensation_sections |
| dtgs |
| highway_gps_coordinates |
| my_assets |
+-----+
4 rows in set (0.00 sec)

mysql> desc dtgs;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id | bigint(20) unsigned | NO | PRI | NULL | auto_increment |
| dtg_model1 | longtext | YES | | NULL | |
| car_number1 | longtext | YES | | NULL | |
| car_type1 | longtext | YES | | NULL | |
| car_registration_number1 | longtext | YES | | NULL | |
| company_registration_number1 | longtext | YES | | NULL | |
| transportation_company_location_code1 | longtext | YES | | NULL | |
| transportation_company_code1 | longtext | YES | | NULL | |
| fuel_code1 | longtext | YES | | NULL | |
| year1 | longtext | YES | | NULL | |
| displacement1 | longtext | YES | | NULL | |
| daily_mileage1 | longtext | YES | | NULL | |
| cumulative_mileage1 | longtext | YES | | NULL | |
| date_information1 | longtext | YES | | NULL | |
| vehicle_velocity1 | int(11) | YES | | NULL | |
| vehicle_rpm1 | int(11) | YES | | NULL | |
| brake_signal1 | tinyint(1) | YES | | NULL | |
| gps_x1 | double | YES | | NULL | |
| gps_y1 | double | YES | | NULL | |
| azimuth1 | longtext | YES | | NULL | |
| acceleration_x1 | double | YES | | NULL | |
| acceleration_y1 | double | YES | | NULL | |
| communication_status_code1 | longtext | YES | | NULL | |
| run_location_code1 | longtext | YES | | NULL | |
| created_at1 | longtext | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
25 rows in set (0.00 sec)
```

그림 70 DATABASE dtgs 생성

2. 서버 S/W와 DB연동

차량관제서버는 디지털운행기록단말기에서 전송된 데이터를 처리하며, 수신된 데이터는 데이터베이스에 연동해서 저장한다. 데이터 값은 차량속도, GPS 위도/경도 좌표, 가속도센서 X/Y 값, TRUE/FALSE 등이 전송된다.

```
getdtg
{"Dtg_model1":"KSD-S+", "Car_number1":"XXXXXXXXBJU608433", "Car_type1":"31", "Car_registration_number1":
"1184363897", "Company_registration_number1":"XXXXX01222", "Transportation_company_location_code1":"t
ransportation_company_location_code", "Transportation_company_code1":"0154267", "Fuel_code1":"fuel_cod
e", "Year1":"year", "Displacement1":"displacement", "Daily_mileage1":"daily_mileage", "Cumulative_mileag
e1":"cumulative_mileage", "Date_information1":"date_information", "Vehicle_velocity1":66, "Vehicle_rpm
1":4000, "Brake_signal1":false, "Gps_x1":36.796324, "Gps_y1":127.167978, "Azimuth1":344.4, "Acceleration
_x1":0.080810546875, "Acceleration_y1":0.01171875, "Communication_status_code1":"communication_status_c
od", "Run_location_code1":"run_location_code", "CreatedAt1":"2022-01-25 10:19:06"
-----DataBase Save Ok-----
```

그림 71 차량관제서버의 송신받은 데이터

ement	daily_mileage	cumulative_mileage	date_information	66	4000	0
0.01171875	communication_status_cod	run_location_code	2022-01-25 10:19:06			

그림 72 데이터베이스의 저장된 데이터

[영상처리 알고리즘개발]

천안삼거리 휴게소 ~ 서울만남의 광장휴게소간 경부고속도로에서 실 주행으로 촬영된 시험영상에서 Ground Truth를 추출하고 해당 영상에서 차선과 실주행 중 확인할 수 있는 안전운행 관련 판단에 대해서 정확도(인식률)과 처리속도를 확인한다.

유로트릭 시뮬레이션 프로그램을 이용하여 다양한 상황에서의 위험운전 시험데이터를 생성하고 해당 영상에서 안전운행 관련 판단에 대해서 정확도(인식률)과 처리속도를 확인한다.

작성된 Ground truth 데이터를 시험대상 시스템이 실주행 영상과 시뮬레이션 영상에 대해서 처리한 결과를 자동 측정 프로그램을 이용하여 정확도(인식률)과 처리속도를 평가한다.

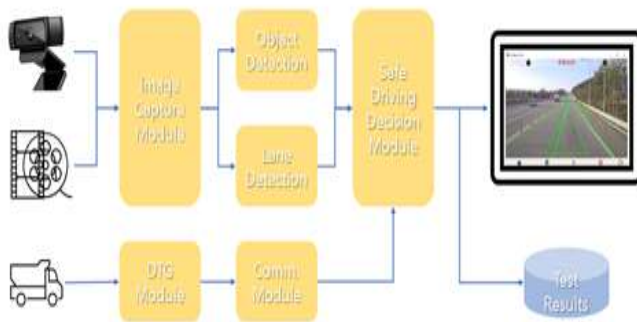


그림 73 안전운행 영상처리 시스템 블록도



그림 74 메인화면

안전운행 영상처리 시스템은 카메라에서 직접 도로영상을 취득하는 경우와 시뮬레이션 모드로 저장된 영상을 불러올 수 있는 2개의 모드로 동작하며, DTG모듈을 통하여 차량의 운행정보를 수신하게 되어있다. 취득 영상은 전방에 있는 차량을 파악하기 위한 객체 검출 모듈과 차선을 검출하기 위한 모듈에 의해서 사용되며, 2개의 inference 모듈이 매 프레임마다 처리한 결과를 안전운행 판단 모듈로 전송하고, 안전운행 판단 모듈은 모든 정보를 활용하여 안전운행 위반 여부를 판단하거나 서버가 판단할 수 있도록 정보를 제공한다.

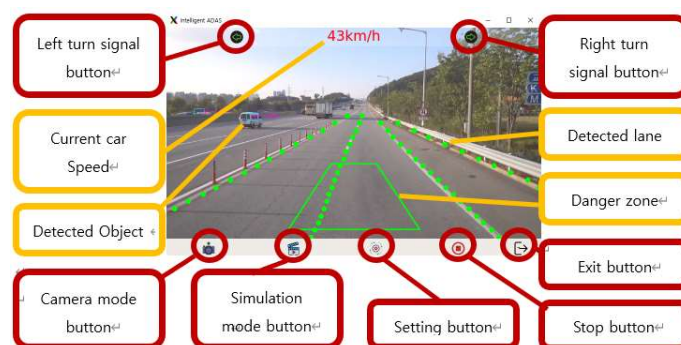


그림 75 프로그램 실행 시 LCD 메인화면

1. 규정차선 및 중앙선 침범 확인기능개발

Camera 및 Simulation mode에서 영상소스가 들어오면 Detection Thread의 Lane Detection에서 차선을 추출 후 CarStatus의 info Control Thread와 CarProp Thread에서 처리해 해당 차량의 주행차선을 판별해 규정차선 기능이 동작한다. 차선을 검출하기 위해 30,000장 도로사진의 차선을 레이블링하여 ResNet-18모델을 이용하여 학습하였다. 중앙선 침범확인기능은 Lane Detection에서 추가로 차선에서 노랑색이 검출되고 차량이 침범했을 경우 기능이 동작한다.

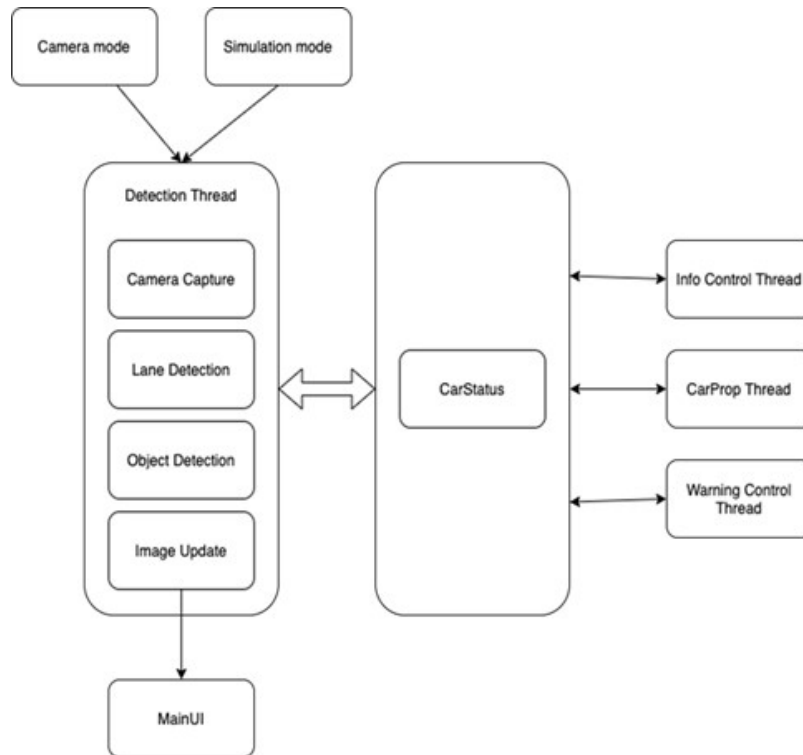


그림 76 영상처리 알고리즘 FLOW CHART

Camera에서 frame을 읽어와서 Lane Detect와 object Detect를 수행한다. 그리고 CarStatus에 저장되어 있는 현재 program setting을 참조하여 setting에 따른 기능을 수행한다. 그 다음 Lane De-parture warning 혹은 Collision warning을 판단 후 연산을 마친 frame을 UI에 새롭게 update를 해준다.

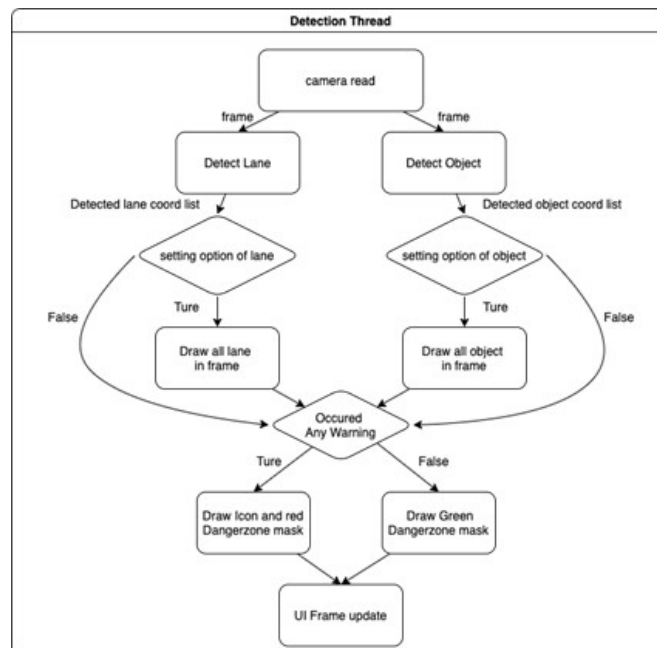


그림 77 Detection Thread

InfoControlThread에서는 CarStatus에 저장되어 있는 Car_speed, Lane coord list, Object coord list를 받아와서 현재 상황에서의 Lane Departure warning과 Collision warning을 판단한다. 그리고 그 결과 값을 다시 CarStatus에 저장해 준다.



CarPropReaderThread에서는 서버와 1초에 한번씩 통신하는 기능을 담당한다. 그래서 서버에서 받아온 data로 CarStatus의 정보를 최신화 해준다. 그리고 그 정보를 바탕으로 현재 안전 운행 판단을 하는 기능을 한다.

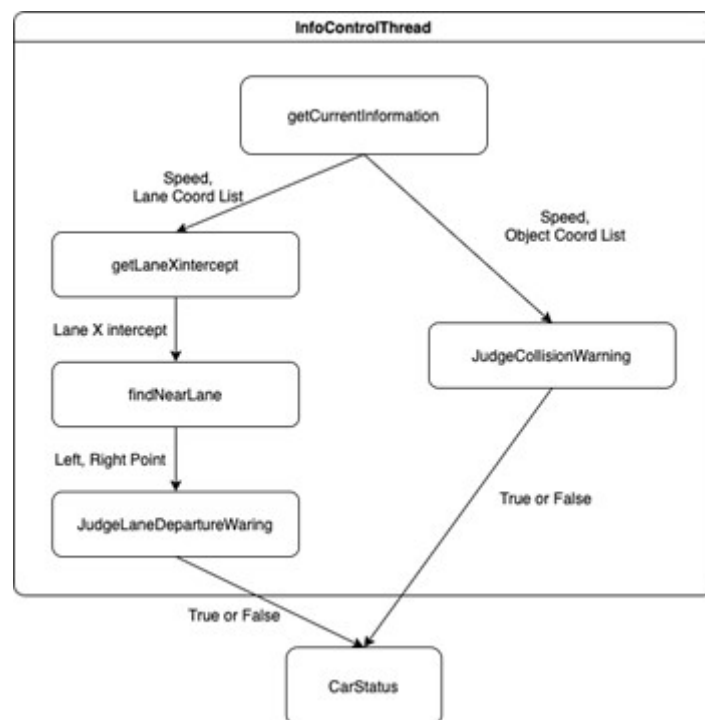


그림 79 InfoControlThread

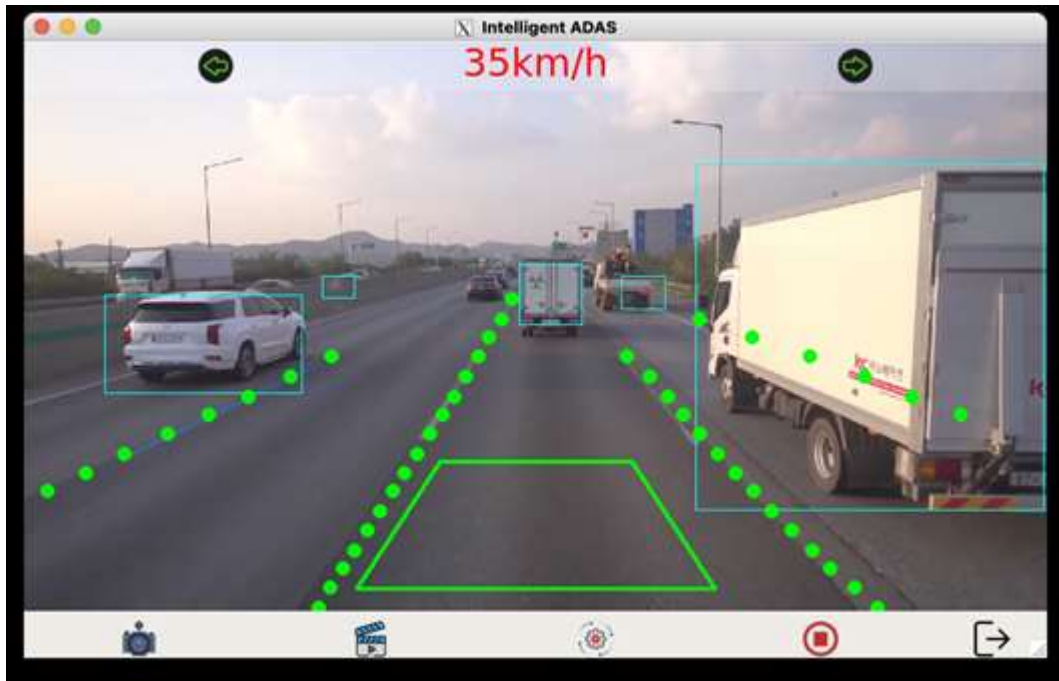


그림 80 규정차선



그림 81 중앙선 침범

2. 불법유턴 확인기능 개발

디지털운행기록단말기에서 취득된 방위각 값을 GRPC프로토콜을 이용하여 ADAS 프로그램에 전송하며, 프로그램은 방위각(속도가 15km/h 이상이고, 8초 안에 좌측 또는 우측160~180° 범위 운행)과 중앙선 침범기능을 활용하여 불법유턴 유무를 확인합니다.



그림 82 불법 유턴

[시스템장치와 서버연동]

1. 통신모듈과 서버 연동

LTE 모뎀을 디지털운행기록단말기에 연결하여 프로그램 실행 시 TCP/IP 프로토콜을 사용하여 관제서버와 연동한다. 서버에서 수신상태일 때 디지털운행기록단말기에서 전송된 데이터를 수신한다.

```

qq@jetson-nano:~$ curl ifconfig.me
223.171.62.116qq@jetson-nano:~$ ping 221.146.63.66
PING 221.146.63.66 (221.146.63.66) 56(84) bytes of data:
64 bytes from 221.146.63.66: icmp_seq=1 ttl=50 time=59.4 ms
64 bytes from 221.146.63.66: icmp_seq=2 ttl=50 time=53.8 ms
64 bytes from 221.146.63.66: icmp_seq=3 ttl=50 time=56.0 ms
64 bytes from 221.146.63.66: icmp_seq=4 ttl=50 time=46.6 ms
64 bytes from 221.146.63.66: icmp_seq=5 ttl=50 time=44.8 ms
64 bytes from 221.146.63.66: icmp_seq=6 ttl=50 time=63.0 ms
64 bytes from 221.146.63.66: icmp_seq=7 ttl=50 time=41.7 ms
64 bytes from 221.146.63.66: icmp_seq=8 ttl=50 time=41.9 ms
64 bytes from 221.146.63.66: icmp_seq=9 ttl=50 time=58.0 ms
64 bytes from 221.146.63.66: icmp_seq=10 ttl=50 time=56.9 ms
64 bytes from 221.146.63.66: icmp_seq=11 ttl=50 time=55.9 ms
64 bytes from 221.146.63.66: icmp_seq=12 ttl=50 time=54.9 ms
64 bytes from 221.146.63.66: icmp_seq=13 ttl=50 time=52.8 ms
64 bytes from 221.146.63.66: icmp_seq=14 ttl=50 time=50.8 ms
64 bytes from 221.146.63.66: icmp_seq=15 ttl=50 time=48.8 ms
64 bytes from 221.146.63.66: icmp_seq=16 ttl=50 time=47.0 ms
64 bytes from 221.146.63.66: icmp_seq=17 ttl=50 time=45.7 ms
64 bytes from 221.146.63.66: icmp_seq=18 ttl=50 time=43.9 ms
64 bytes from 221.146.63.66: icmp_seq=19 ttl=50 time=42.9 ms
64 bytes from 221.146.63.66: icmp_seq=20 ttl=50 time=41.8 ms
64 bytes from 221.146.63.66: icmp_seq=21 ttl=50 time=39.8 ms
64 bytes from 221.146.63.66: icmp_seq=22 ttl=50 time=58.0 ms
64 bytes from 221.146.63.66: icmp_seq=23 ttl=50 time=57.0 ms
64 bytes from 221.146.63.66: icmp_seq=24 ttl=50 time=55.7 ms
64 bytes from 221.146.63.66: icmp_seq=25 ttl=50 time=53.8 ms
64 bytes from 221.146.63.66: icmp_seq=26 ttl=50 time=53.9 ms
64 bytes from 221.146.63.66: icmp_seq=27 ttl=50 time=50.9 ms
64 bytes from 221.146.63.66: icmp_seq=28 ttl=50 time=48.9 ms
64 bytes from 221.146.63.66: icmp_seq=29 ttl=50 time=48.0 ms
64 bytes from 221.146.63.66: icmp_seq=30 ttl=50 time=46.9 ms
  
```

그림 83 TCP/IP 프로토콜에서 ICMP로 통신

```
--- 221.146.63.66 ping statistics ---
100 packets transmitted, 100 received, 0% packet loss, time 99145ms
rtt min/avg/max/mdev = 33.572/51.579/68.022/6.417 ms
qq@jetson-nano:~$
```

그림 84 TCP/IP 통신결과

```
100 333.67 0.25512802125 0.0519109075 communication_status_cod run_location_code 2022-02-05 17:13:46}
2022/02/05 17:13:47 Server send : 0 0 0 0
4(KSD-S- XXXXXX030U068431 31 1184361897 XXXXX01222 transportation_company_location_code 0154267 fuel_code year displacement daily_mileage cumulative_mileage date_information 52 4000 false 36.790185 127.17
1119 333.58 0.263125 -0.0307431975 communication_status_cod run_location_code 2022-02-05 17:13:47}
2022/02/05 17:13:48 Server send : 0 0 0 0
4(KSD-S- XXXXXX030U068431 31 1184361897 XXXXX01222 transportation_company_location_code 0154267 fuel_code year displacement daily_mileage cumulative_mileage date_information 54 4000 false 36.790318 127.17
1238 334.56 0.19021875 0.03784176875 communication_status_cod run_location_code 2022-02-05 17:13:48}
2022/02/05 17:13:48 Server send : 0 0 0 0
4(KSD-S- XXXXXX030U068431 31 1184361897 XXXXX01222 transportation_company_location_code 0154267 fuel_code year displacement daily_mileage cumulative_mileage date_information 55 4000 false 36.790456 127.17
116 335.18 0.2410802125 0.0343796875 communication_status_cod run_location_code 2022-02-05 17:13:48}
2022/02/05 17:13:49 Server send : 0 0 0 0
4(KSD-S- XXXXXX030U068431 31 1184361897 XXXXX01222 transportation_company_location_code 0154267 fuel_code year displacement daily_mileage cumulative_mileage date_information 57 4000 false 36.790595 127.17
1875 334.23 0.13427734375 0.054931640625 communication_status_cod run_location_code 2022-02-05 17:13:50}
2022/02/05 17:13:50 Server send : 0 0 0 0
4(KSD-S- XXXXXX030U068431 31 1184361897 XXXXX01222 transportation_company_location_code 0154267 fuel_code year displacement daily_mileage cumulative_mileage date_information 57 4000 false 36.790736 127.17
4009 333.82 0.2657184375 -0.0094082125 communication_status_cod run_location_code 2022-02-05 17:13:50}
2022/02/05 17:13:51 Server send : 0 0 0 0
4(KSD-S- XXXXXX030U068431 31 1184361897 XXXXX01222 transportation_company_location_code 0154267 fuel_code year displacement daily_mileage cumulative_mileage date_information 57 4000 false 36.790877 127.17
4004 333.83 0.26393128125 0.026611328125 communication_status_cod run_location_code 2022-02-05 17:13:51}
2022/02/05 17:13:52 Server send : 0 0 0 0
4(KSD-S- XXXXXX030U068431 31 1184361897 XXXXX01222 transportation_company_location_code 0154267 fuel_code year displacement daily_mileage cumulative_mileage date_information 57 4000 false 36.791017 127.17
4022 334.44 0.220458984375 -0.0512225625 communication_status_cod run_location_code 2022-02-05 17:13:54}
2022/02/05 17:13:54 Server send : 0 0 0 0
4(KSD-S- XXXXXX030U068431 31 1184361897 XXXXX01222 transportation_company_location_code 0154267 fuel_code year displacement daily_mileage cumulative_mileage date_information 57 4000 false 36.791155 127.17
4728 333.54 0.18020984375 0.06225989375 communication_status_cod run_location_code 2022-02-05 17:13:54}
2022/02/05 17:13:55 Server send : 0 0 0 0
4(KSD-S- XXXXXX030U068431 31 1184361897 XXXXX01222 transportation_company_location_code 0154267 fuel_code year displacement daily_mileage cumulative_mileage date_information 56 4000 false 36.791293 127.17
4652 333.59 0.179931640625 0.04467774375 communication_status_cod run_location_code 2022-02-05 17:13:55}
2022/02/05 17:13:55 Server send : 0 0 0 0
4(KSD-S- XXXXXX030U068431 31 1184361897 XXXXX01222 transportation_company_location_code 0154267 fuel_code year displacement daily_mileage cumulative_mileage date_information 56 4000 false 36.791431 127.17
4679 333.06 0.21925094375 -0.03092574375 communication_status_cod run_location_code 2022-02-05 17:13:55}
2022/02/05 17:13:57 Server send : 0 0 0 0
4(KSD-S- XXXXXX030U068431 31 1184361897 XXXXX01222 transportation_company_location_code 0154267 fuel_code year displacement daily_mileage cumulative_mileage date_information 55 4000 false 36.791568 127.17
4649 334.48 0.17431040625 0.02026361875 communication_status_cod run_location_code 2022-02-05 17:13:58}
2022/02/05 17:13:59 Server send : 0 0 0 0
4(KSD-S- XXXXXX030U068431 31 1184361897 XXXXX01222 transportation_company_location_code 0154267 fuel_code year displacement daily_mileage cumulative_mileage date_information 56 4000 false 36.791707 127.17
4609 334.79 0.180625 0.048421875 communication_status_cod run_location_code 2022-02-05 17:13:59}
2022/02/05 17:14:00 Server send : 0 0 0 0
4(KSD-S- XXXXXX030U068431 31 1184361897 XXXXX01222 transportation_company_location_code 0154267 fuel_code year displacement daily_mileage cumulative_mileage date_information 56 4000 false 36.791848 127.17
4277 334.58 0.268009375 -0.01393601875 communication_status_cod run_location_code 2022-02-05 17:14:01}
2022/02/05 17:14:01 Server send : 0 0 0 0
4(KSD-S- XXXXXX030U068431 31 1184361897 XXXXX01222 transportation_company_location_code 0154267 fuel_code year displacement daily_mileage cumulative_mileage date_information 55 4000 false 36.791984 127.17
4246 334.64 0.135009765625 0.00439451125 communication_status_cod run_location_code 2022-02-05 17:14:01}
2022/02/05 17:14:02 Server send : 0 0 0 0
4(KSD-S- XXXXXX030U068431 31 1184361897 XXXXX01222 transportation_company_location_code 0154267 fuel_code year displacement daily_mileage cumulative_mileage date_information 55 4000 false 36.792121 127.17
4106 334.06 0.2584176875 -0.00239705625 communication_status_cod run_location_code 2022-02-05 17:14:03}
2022/02/05 17:14:05 Server send : 0 0 0 0
4(KSD-S- XXXXXX030U068431 31 1184361897 XXXXX01222 transportation_company_location_code 0154267 fuel_code year displacement daily_mileage cumulative_mileage date_information 55 4000 false 36.792258 127.17
4087 334.83 0.22242109375 0.02685546875 communication_status_cod run_location_code 2022-02-05 17:14:05}
2022/02/05 17:14:06 Server send : 0 0 0 0
4(KSD-S- XXXXXX030U068431 31 1184361897 XXXXX01222 transportation_company_location_code 0154267 fuel_code year displacement daily_mileage cumulative_mileage date_information 55 4000 false 36.792394 127.17
4011 335.43 0.2231976875 -0.00972421875 communication_status_cod run_location_code 2022-02-05 17:14:06}
2022/02/05 17:14:07 Server send : 0 0 0 0
4(KSD-S- XXXXXX030U068431 31 1184361897 XXXXX01222 transportation_company_location_code 0154267 fuel_code year displacement daily_mileage cumulative_mileage date_information 55 4000 false 36.79253 127.169
395 335.84 0.11840820125 0.0308625 communication_status_cod run_location_code 2022-02-05 17:14:07}
2022/02/05 17:14:08 Server send : 0 0 0 0
```

그림 85 디지털운행기록단말기에서 서버-데이터수신완료 메시지 출력

2. 차량운행기록장치 연동

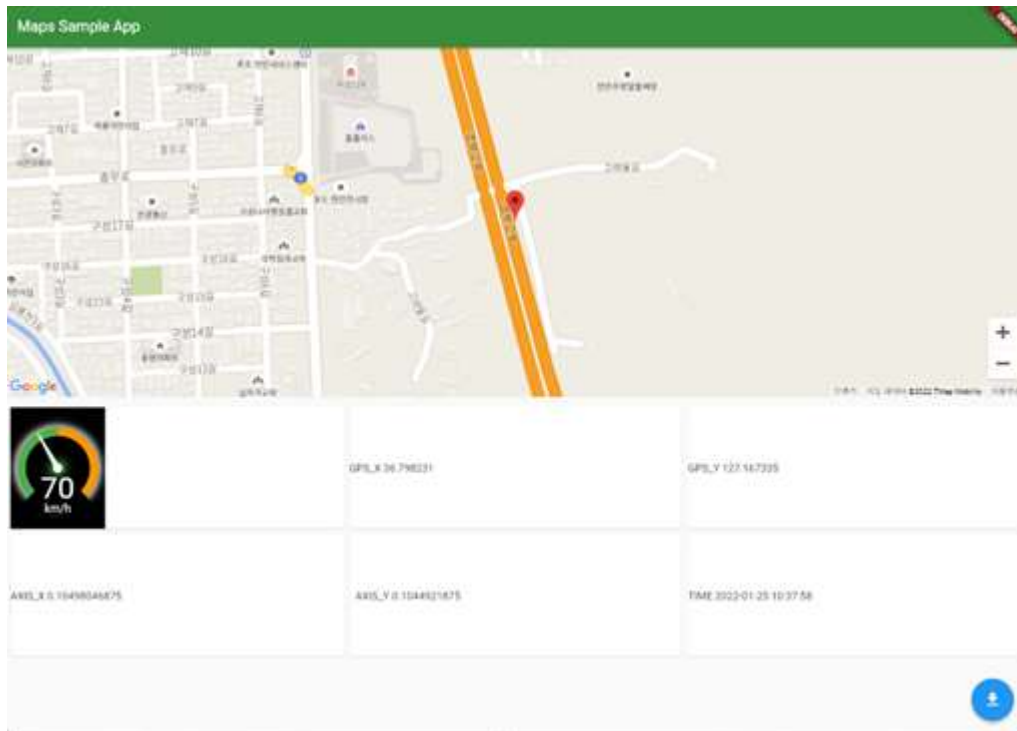
디지털운행기록단말기에서 1초 단위로 수집된 데이터를 .txt포맷으로 저장하고 LTE모뎀을 통해서 차량관제 서버로 전송하며, 차량관제서버는 .txt데이터 내용과 동일하게 데이터베이스에 연동하여 데이터를 저장한다.

date	
2021-09-14	2021-12-01
2021-09-17	2021-12-02
2021-10-13	2021-12-03
2021-10-14	2021-12-04
2021-10-15	2021-12-06
NULL	2021-12-07
2021-10-18	2021-12-08
2021-10-19	2021-12-11
2021-10-20	2021-12-12
2021-10-21	2021-12-13
2021-10-25	2021-12-15
2021-10-26	2021-12-21
2021-10-27	2021-12-22
2021-10-28	2021-12-23
2021-10-29	2021-12-30
2021-11-01	2021-12-31
2021-11-02	2022-01-01
2021-11-03	2022-01-02
2021-11-06	2022-01-03
2021-11-07	2022-01-04
2021-11-08	2022-01-05
2021-11-09	2022-01-06
2021-11-10	2022-01-07
2021-11-11	2022-01-08
2021-11-12	2022-01-10
2021-11-13	2022-01-11
2021-11-15	2022-01-12
2021-11-16	2022-01-14
2021-11-17	2022-01-17
2021-11-18	2022-01-18
2021-11-19	2022-01-19
2021-11-21	2022-01-20
2021-11-22	2022-01-21
2021-11-25	2022-01-24
2021-11-26	2022-01-25

그림 86 생성된 데이터 총 68개
(2021-09-14 ~ 2022-01-25 기간)

3. 애플리케이션 서버연동

웹 브라우저(크롬, 사파리, 엣지 등)에서 디지털운행기록단말기에서 차량관제서버로 전송된 데이터가 화면에 속도, RPM, 가속도센서, GPS 등 데이터를 표출해준다.



4. 모바일 Dapp 연동

모바일 화면에서 디지털운행기록단말기에서 차량관제서버로 전송된 데이터가 화면에 속도, RPM, 가속도센서, GPS 등 데이터를 표출해준다.



[시제품 테스트]

1. 유사환경하에서 동작확인

천안삼거리 휴게소 ~ 서울만남의 광장휴게소 구간의 실 주행 데이터와 영상을 저장하여 시뮬레이션에서 다시 동작하여 프로그램이 동작하는지 확인한다. 추가로 고속도로에서 테스트하기 힘든 중앙선 침범, 불법유턴, 급차로 변경 등 실주행에서 위험한 행동은 유로트럭3으로 유사환경을 맞추고 동작하는지 확인하였다.

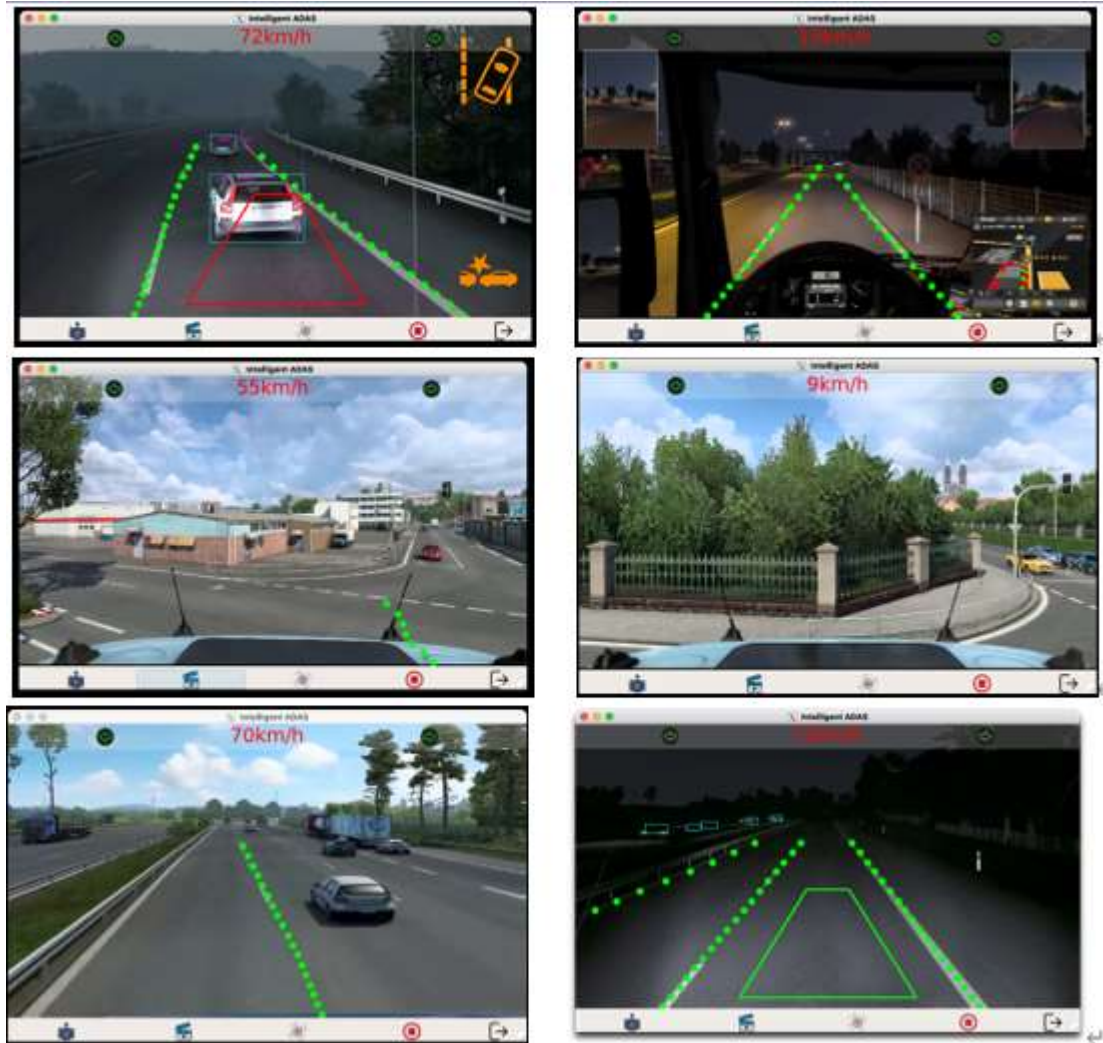


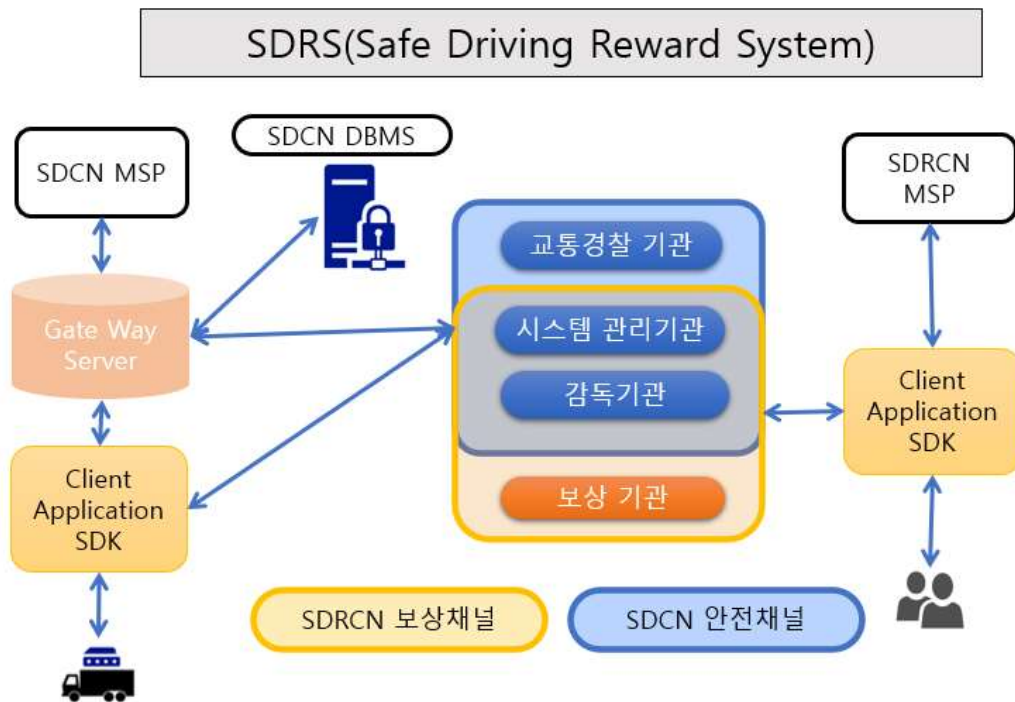
그림 89 유로트럭3 환경에서 기능동작

[블록체인 보상시스템]

1.블록체인 사고안전 보상시스템 통합

1.1 블록체인 사고안전 보상시스템(SDRS)의 개요

사고안전 보상 시스템(SDRS:Safe Driving Reward System)은 하이퍼레저패브릭 기반의 블록체인 네트워크이다. 하이퍼레저패브릭 블록체인 네트워크에서는 채널단위로 원장을 공유하게 되는데 같은 채널에 포함된 조직과 참여자는 원장의 데이터 읽기, 쓰기 등의 접근권한을 관리할 수 있다. 2차년도 연구결과 사고안전 블록체인 네트워크를 개발하였으며, 3 차 년도 연구결과 사고안전 보상 블록체인 네트워크를 개발하였다. 그 결과 두 시스템을 통합하여 블록체인 사고안전 보상시스템을 개발 통합 하였다. 시스템의 구성은 하이퍼레저패브릭의 채널기능을 이용하여 안전채널(SDCN:Safe Driving Channel Network) 과 보상채널(SDRCN:Safe Driving Reward Channel Network)로 구현 하였다. 하나의 채널에만 소속되어 있는 기관이 있어 다른 채널의 원장에 접근할 수 없으며, 두 개의 채널 모두에 소속되어 있는 기관은 양쪽 채널의 데이터에 접근할 수 있다. 그림 1과 같이 안전채널에서는 차량의 운행정보와 안전운전 행위에 대한 데이터를 안전채널에 기록 관리하며, 보상채널은 안전채널 정보를 바탕으로 안전운전 행위에 대한 금전적 보상정보를 장부에 기록 관리하고 있다.



(그림 90) 블록체인 기반 사고안전 보상시스템 구성

채널별 참여기관은 표 1과 같이 안전채널은 운영기관, 감독기관, 교통경찰 기관이 시스템운영에 참여하며, 보상채널은 운영기관 감독기관 보상기관 등이 참여할 수 있다.

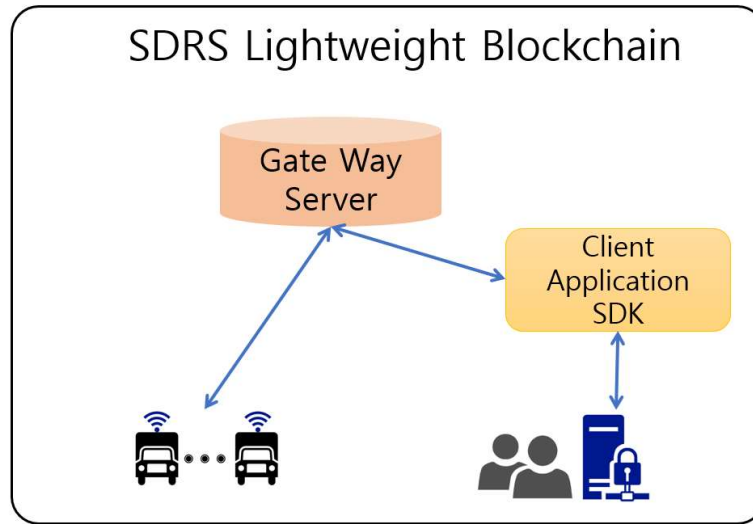
(표 21) 안전채널(SDCN)과 보상채널(SDRCN)

참여기관	SDCN 안전 채널	SDRCN 보상 채널
운영기관	○	○
감독기관	○	○
교통경찰	○	
보상기관		○

운영기관은 시스템 유지보수를 담당하며, 감독기관은 사고안전 보상시스템이 정상적으로 운용되는지를 관리 감독하고, 데이터의 입출력을 담당하기 위하여 안전채널 및 보상채널에 모두 접근할 수 있다. 교통경찰의 경우 화물차의 안전운전을 지원할 수 있도록 안전채널의 정보를 읽을 수 있는 권한을 가지며, 보상기관은 보상에 관련된 참여자의 보상채널 데이터에 접근을 관리하기 위한 기관이다.

1.2 경량 블록체인 시스템 적용

본 연구에서 개발된 경량 블록체인 시스템은 온, 오프라인 모두에서 동작할 수 있으며, 모바일 기기에 적합한 특징을 가지고 있다. 차량운행 중 네트워크 접속이 단절된 경우 서버와 단말기 간의 상호인증과정을 수행할 수 없게 된다. 이 경우 서버에서 제공한 랜덤시드(Random Seed)를 활용하여 오프라인에서도 안전하게 사용할 수 있는 알고리즘을 개발 적용하였다.



(그림 91) 경량 블록체인 네트워크

랜덤시드를 활용한 경량 블록체인 시스템은 폐쇄형 블록체인 시스템이나 개방형 블록체인 시스템과 블록체인의 구조에서 차이를 보인다.

Public Blockchain	Private Blockchain	Light Weight Blockchain
Previous Hash	Previous Hash	Previous Hash
Mercle Root Hash	Transaction Endorsement	Data File Hash
Time Stamp	...	Random Hash
Nonce	Transaction Endorsement	
Block Hash	Block Hash	Block Hash

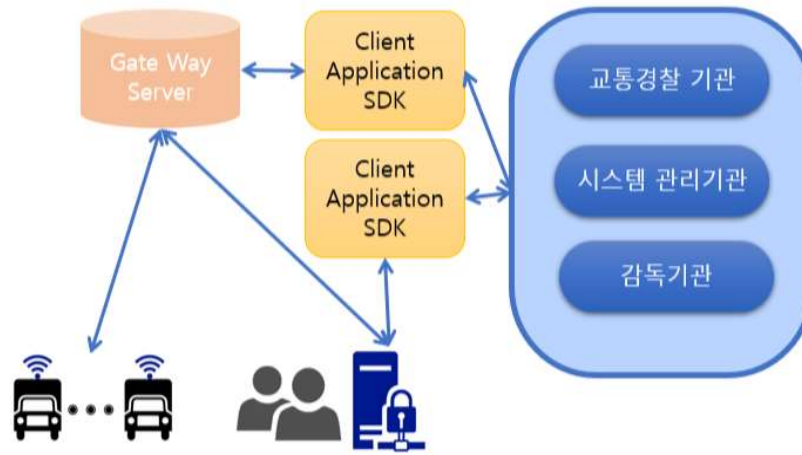
(그림 92) 블록체인 구조의 비교

개방형 블록체인의 블록구조는 그림 3과 같이 Nonce를 이용한 작업증명으로 위변조방지 기능을 수행하며, 폐쇄형 블록체인 시스템은 합의 과정에 참여한 피어노드(Peer Node)의 서명을 이용하여 위변조 방지기능을 수행한다. 경량 블록체인에서는 서버에서 제공된 랜덤시드를 이용하여 통신이 끊긴 상태에서도 위변조방지가 가능하게 한다.

1.3 안전채널(SDCN)

1.3.1 안전채널(SDCN)의 개요

안전채널은 운행 중 위험 구간이나 눈, 비, 안개 등의 상황에서 안전운전을 수행한 보상을 제공할 목적으로 마일리지 포인트를 제공하거나 과속, 차간거리 미확보 등 난폭운전을 한 경우 이에 대한 불이익을 주어 교통사고를 예방하고 안전운전을 습관화할 수 있도록 지원하는 시스템이다. 그림 4와 같이 차량에 설치된 단말기에는 온라인 또는 오프라인 상태에서 안전운전에 관련된 데이터를 저장하고 위변조 방지를 위한 경량 블록체인이 적용된다.



(그림 93) 안전채널 네트워크

단말기에 기록된 데이터는 게이트웨이를 통하여 SDCN 데이터베이스에 등록된다. 데이터베이스에 사고안전 데이터가 등록되면 분산응용프로그램(DAPP)을 통하여 안전채널 블록체인에 데이터 셋을 기록하게 된다. 또한, 관리자나 안전채널에 등록된 사용자는 응용프로그램을 통해 체인코드를 실행하여 블록체인 장부를 검색하거나 변경할 수 있다.

1.3.2 안전채널(SDCN) 데이터 셋

안전채널의 데이터 셋은 차량이 운전 중에 발생하는 안전운전에 관련된 이벤트를 기록한다. 하이퍼레저 패브릭 블록체인의 데이터기록 방법은 키(key)와 값(Value)의 쌍을 갖는 상태 데이터베이스(World State Database)에 키와 값을 기록 변경하여 현 상태 값을 유지하도록 구성되어 있다. 상태 데이터베이스는 현재의 상태만을 기록하고 중복된 데이터가 입력되면 이전의 상태는 블록에 기록하고 변경된 값만 상태 데이터베이스에 기록한다.

하이퍼레저 패브릭에서 체인코드(Smart Contract)는 블록체인에 데이터를 입출력할 수 있는 인터페이스를 제공하며, 데이터 셋에 접속 가능한 체인코드를 개발하여 피어노드(Peer Node)에 설치하여야 한다. 표 2는 안전채널에서 사용하는 데이터 셋의 구성이다.

dtgNo는 일련번호로 키를 나타내며, eventDate (언제), GPSx, GPSy(어디서), eventID(어떤) 이벤트가 발생했는지의 정보를 나타낸다. pointValue (마일리지) 값은 안전운전에 도움이 되는 이벤트는 양의 값, 위험운전에 해당하면 음의 값을 주며, 마일리지를 보상금액으로 환산하기위한 변환율 cRate를 사용하였다.

(표 22) 안전채널 데이터셋

항목	내용
dtgNo	일련번호
driverID	운전자 번호
GPSx	위치정보 위도
GPSy	위치정보 경도
roadID	도로번호
eventID	이벤트 코드
pointValue	마일리지 포인트
cRate	환산값
eventDate	이벤트 발생 시간

블록체인 네트워크에 체인코드가 설치되면 명령입력 인터페이스(CLI:Command Line Interface)를 통하여 체인코드를 호출하거나, 분산응용프로그램(DAPP)을 개발하여 체인코드를 호출한다. 본 연구에서는 분산응용프로그램을 개발하여 적용하였다.

1.3.3 안전채널(SDCN) 분산응용프로그램(DAPP)

체인코드를 호출하기 위한 분산응용프로그램의 함수들과 기능은 표 3과 같다. 함수명칭의 명명 방법으로는 T로 시작하는 함수는 순차 블록처리 능력을 측정하는 측정함수이며, Dtg는 안전채널에서 사용되는 함수를 나타낸다. Creat, Read, Update, Delete 명칭은 블록의 생성, 읽기, 수정, 삭제를 의미한다. 접두어로 사용된 Ccm은 병렬처리 함수를 나타내며, 접미어로 Asset은 블록체인이 운용 중에 사용하는 함수이다. History는 특정블록의 변화과정 즉, 생성, 변경, 삭제의 모든 과정을 검색하는 함수이며, ByDriverID는 특정운전자에 해당하는 블록을 호출하는 함수이다.

(표 23) 안전채널 분산응용 프로그램의 함수

함수명칭	기능
TDtgCreate	순차 블록생성 성능측정함수
TDtgRead	순차 블록읽기 성능측정함수
TDtgUpdate	순차 블록수정 성능측정함수
TDtgDelete	순차 블록삭제 성능측정함수
CcmDtgCreate	병렬처리 블록생성 성능측정함수
CcmDtgRead	병렬처리 블록읽기 성능측정함수
CcmDtgUpdate	병렬처리 블록수정 성능측정함수
CcmDtgDelete	병렬처리 블록삭제 성능측정함수
CcmDtgCreateAsset	병렬처리 블록생성 함수
CcmDtgReadAsset	병렬처리 블록읽기 함수
CcmDtgUpdateAsset	병렬처리 블록수정 함수
CcmDtgDeleteAsset	병렬처리 블록삭제 함수
DtgHistoryAsset	키값에 해당하는 모든 블록정보 읽기 함수
DtgReadByDriverIDAsset	운전자별 블록정보 읽기 함수

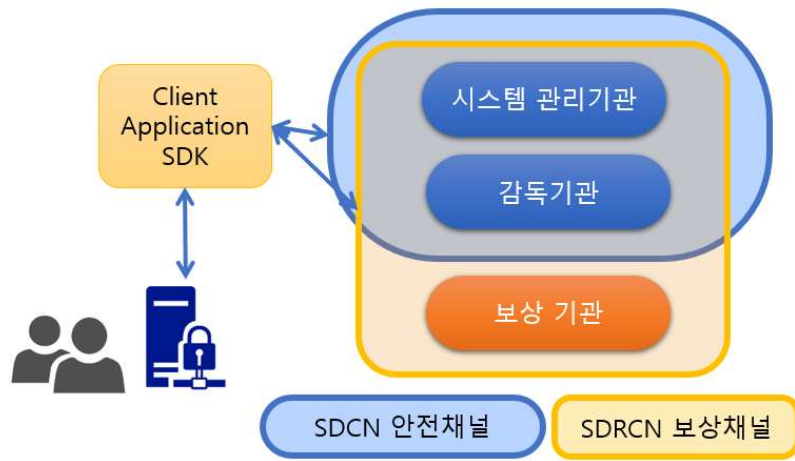
TDtgCreate는 순차 블록생성 성능측정 함수로 체인코드를 순차적으로 호출하여 실행될 때마다 하나의 블록을 생성하고 블록이 완성되면 다음 블록을 생성하도록 하는 성능측정 함수다.

순차블록 생성의 특성상 블록당 한 개의 메시지만 담고 있어 블록처리 성능이 낮은 결과를 나타낸다. TDtgRead, TDtgUpdate, TDtgDelete 함수는 순차읽기, 순차수정, 순차삭제를 수행하여 하나의 블록을 처리한 후 다음 블록을 순차적으로 처리하여 성능을 측정하는 함수다. CcmDtgCreate, CcmDtgRead, CcmDtgUpdate, CcmDtgDelete 함수는 이름에서 알 수 있듯이 병렬처리로 블록을 처리함으로써 성능을 향상시킨 함수다. DtgHistoryAsset 함수는 key값에 해당하는 모든 블록을 반환하는 함수이다. 즉 블록의 모든 변경이력이 출력된다. 만약 블록이 삭제된 경우 블록의 모든 필드값은 리셋되며, 블록삭제 상태를 나타내는 논리값 FALSE(0)를 출력한다. 운전자별 블록정보 읽기 함수 DtgReadByDriverIDAsset은 상태 데이터베이스에서 운전자와 관련된 데이터를 출력하는 함수로 운전자의 정보 검색에 사용된다.

1.4 보상채널(SDRCN)

1.4.1 보상채널(SDRCN)의 개요

보상채널은 안전채널에 기록된 마일리지 포인트를 이용하여 보상을 제공한다. 따라서 이 정보를 바탕으로 차량을 운행한 운전자에게 보상이 주어지며, 주어진 보상은 운전자별로 포인트가 누적된다.



(그림 94) 보상채널 네트워크

관리기관의 경우 보상채널과 안전채널의 데이터에 접근할 수 있으며 안전채널에 기록된 장부를 열람하여 운전자별 보상 포인트를 보상 네트워크 장부에 기록하도록 한다. 기록된 보상 포인트는 증감 정보를 장부에 기록하도록 구성되었다.

1.4.2 보상채널(SDRCN) 데이터 셋

보상채널의 데이터 셋은 차량 운행 중에 발생하는 안전운전에 관련된 이벤트가 기록된 안전채널의 정보를 읽고 보상 포인트로 환산하여 보상채널에 기록되는 값이다. 보상 포인트 값은 증감 값으로 표현되며, 안전운전 이벤트의 경우는 양의 값이 되고, 위험운전 이벤트의 경우 음의 값이 된다. 이때 사용되는 보상 네트워크의 데이터 셋은 표 4와 같다.

(표 24) 보상채널 데이터 셋

항목	내용
dtgNo	일련번호
driverID	운전자 번호
eventID	이벤트 코드
eventDate	이벤트 발생 일시
dtgValue	보상 포인트 증감값

보상채널에 참가한 기관이나 사용자는 데이터 셋에 접근할 수 있는 분산응용프로그램의 함수를 이용하여 블록을 생성, 변경, 검색, 삭제를 수행한다.

1.4.3 보상채널(SDRCN) 분산응용프로그램

체인코드를 호출하기 위한 분산응용프로그램의 함수들과 기능은 표 5와 같다.

(표 25) 보상채널 분산응용 프로그램의 함수

함수명칭	기능
TCompCreate	순차 블록생성 성능측정함수
TCompRead	순차 블록읽기 성능측정함수
TCompUpdate	순차 블록수정 성능측정함수
TCompDelete	순차 블록삭제 성능측정함수
CcmCompCreate	병렬처리 블록생성 성능측정함수
CcmCompRead	병렬처리 블록읽기 성능측정함수
CcmCompUpdate	병렬처리 블록수정 성능측정함수
CcmCompDelete	병렬처리 블록삭제 성능측정함수
CcmCompCreateAsset	병렬처리 블록생성 함수
CcmCompReadAsset	병렬처리 블록읽기 함수
CcmCompUpdateAsset	병렬처리 블록수정 함수
CcmCompDeleteAsset	병렬처리 블록삭제 함수
CcmCompHistoryAsset	킷값에 해당하는 모든 블록정보 읽기 함수
CcmCompReadByDriverIDAsset	운전자별 블록정보 읽기 함수
CcmBanlanceQueryAsset	운전자별 보상잔액 검색
CcmBalanceHistoryAsset	운전자별 잔액 변동내역 검색

안전채널 함수에 추가된 것은 보상 잔액을 검색하는 CcmBanlanceQueryAsset 함수와 운전자별 보상 잔액의 변동 내역을 보여주는 CcmBanlanceHistoryAsset 함수이다.

1.5 사고안전시스템(SDRS)

1.5.1 블록체인 네트워크 구성

사고안전시스템 블록체인 네트워크 시스템의 동작을 확인하기 위하여 표 6과 같이 2개의 조직과 조직별 2개의 피어노드, 블록을 생성하고 합의과정을 수행하는 오더러노드(Orderer Node)로 구성하였다.

(표 26) 노드 수준에서 조직의 구성

노드	조직명	도메인
Orderer	Orderer	orderer.example.com
peer0	org1	peer0.org1.example.com
peer1	org1	peer1.org1.example.com
peer0	org2	peer0.org2.example.com
peer1	org2	peer1.org2.example.com

1.5.2 하드웨어 구성

하이퍼레저 패브릭 블록체인 시스템은 도커컴포즈 (docker compose)를 이용하여 가상화 기반(Virtual Machine Based)시스템으로 여러 대의 가상서버를 구성하였다. 본 연구에서 사용한 네트워크의 경우 피어노드마다 서버가 운영되어야 하며 CouchDB를 사용하는 경우 피어노드마다 데이터베이스 가상서버가 구동된다. 따라서 고성능서버의 사용이 블록체인의 성능을 높이는데 직접적인 관계가 있다. 본 연구에서 사용된 시스템의 사양을 표 7에 나타내었다.

(표 27) 시스템 사양

구 분	사 양
CPU	Intel Core i9-990xe @3.0GHZ x36 core
GPU	Nvidia 1080 8GB memory
Memory	64G DDR4 2133MHZ
Storage SSD	1TB SAMSUNG 970 PRO
Hard disk	3TB TOSHIBA 7200RPM SATA 3.0 6Gb/S
Network	Ethernet 802.3bz

1.5.3 소프트웨어 구성

본 연구에서 구축한 블록체인 시스템은 높은 데이터처리 능력을 요구하므로 고성능의 블록체인 네트워크 구성을 목적으로 하여 개발언어를 선정하였다. 일반적으로 하이퍼레저 패브릭에서 체인코드개발에 사용되는 언어는 Go, Java, Node.js가 사용되며, 분산응용프로그램 개발은 Go, Python, Java 버전의 SDK가 제공된다. 개발언어별 특징으로 인하여 블록체인 처리속도에 차이가 있으며, Go언어가 가장 성능이 우수한 것으로 알려졌다. IBM Blockchain Platform의 Hyperledger Blockchain Performance Reports에 따르면 Node.js 165TPS, Go언어 860TPS를 나타내고 있다. 본 연구에서는 Go언어를 체인코드 개발언어로 선정하였으며, 분산응용프로그램 개발에도 Go언어를 사용하여 개발하였다. 하이퍼레저 패브릭 블록체인 개발을 위해 설치해야 하는 프로그램과 버전정보를 표 8에 나타내었다.

(표 28) 소프트웨어 구성

구 분	사 양
Operating System	Ubuntu 18.04 LST
fabric-contract-api-go	Version v1.1.1
Smart Contract Language	Go 1.6
Hyperledger Fabric Client golang SDK	V1.0
DAPP Language	Go 1.6
Remote Desktop S/W	Anydesk 6.3

2 블록체인 사고안전 보상시스템(SDRS) 성능평가

2.1 안전채널(SDCN) 성능평가 상세내용

표 29 성능평가 명령어 사용방법

네트워크	시험항목	실행 명령
사고안전	블록생성	./dapp CcmDtgCreate 0 500 100
	블록읽기	./dapp CcmDtgRead 0 500 100
	블록변경	./dapp CcmDtgUpdate 0 500 100
	블록삭제	./dapp CcmDtgDelete 0 500 100

2.1.1블록 생성 명령 설명

./dapp : 사고안전 및 사고안전 보상 블록체인 네트워크 접속프로그램

CcmDgrCreate : 블록생성 체인코드를 실행하는 함수

0 블록생성시작번호

500 : 블록생성시간을 측정하기위한 블록의 수로서 500개의 블록을 생성하고 평균블록시간을 측정

100 : 500개의 블록을 생성한 후 계산된 평균블록생성시간을 100회 반복하여 수행

결과값 출력 총 10000개의 블록을 생성하고 결과값으로 TPS(Transaction per second)값을 출력한다.

2.1.2블록 읽기 명령 설명

./dapp : 사고안전 및 사고안전 보상 블록체인 네트워크 접속프로그램

CcmDtgRead : 블록읽기 체인코드를 실행하는 함수

0: 블록처리 시작번호

500: 블록처리 시간을 측정하기위해 한번에 처리하는 블록의 수로서 500개의 블록을 처리하고 평균블록 처리 시간을 측정

100 : 500개의 블록을 처리한 후 계산된 평균블록 처리시간을 100회 반복하여 수행

결과값 출력: 총 10000개의 블록을 처리하고 결과값으로 TPS(Transaction per second)값을 출력한다.

2.1.3블록 변경 명령 설명

./dapp : 사고안전 및 사고안전 보상 블록체인 네트워크 접속프로그램

CcmDtgUpdate : 블록변경 체인코드를 실행하는 함수

0: 블록처리 시작번호

500: 블록처리 시간을 측정하기위해 한번에 처리하는 블록의 수로서 500개의 블록을 처리하고 평균 블록 처리 시간을 측정

100 : 500개의 블록을 처리한 후 계산된 평균블럭 처리시간을 100회 반복하여 수행

결과값 출력: 총 10000개의 블록을 처리하고 결과값으로 TPS(Transaction per second)값을 출력한다.

2.1.4블록삭제 명령 설명

./dapp : 사고안전 및 사고안전 보상 블록체인 네트워크 접속프로그램

CcmDtgDelete : 블록삭제 체인코드를 실행하는 함수

0: 블록처리 시작번호

500: 블록처리 시간을 측정하기 위해 한번에 처리하는 블록의 수로서 500개의 블록을 처리하고 평균 블록처리 시간을 측정

100 : 500개의 블록을 처리한 후 계산된 평균블럭 처리시간을 100회 반복하여 수행

결과값 출력: 총 10000개의 블록을 처리하고 결과값으로 TPS(Transaction per second)값을 출력한다.

2.2 보상채널(SDRCN) 성능평가 상세내용

표 30 시험평가 명령어 사용법

네트워크	시험항목	실행 명령
사고안전 보상	블록생성	./dapp CcmCompCreate 0 500 100
	블록읽기	./dapp CcmCompRead 0 500 100
	블록변경	./dapp CcmCompUpdate 0 500 100
	블록삭제	./dapp CcmCompDelete 0 500 100

2.2.1블록 생성 명령 설명

./dapp : 사고안전 및 사고안전 보상 블록체인 네트워크 접속프로그램

CcmCompCreate : 블록생성 체인코드를 실행하는 함수

0 블록생성시작번호

500 : 블록생성시간을 측정하기위한 블록의 수로서 500개의 블록을 생성하고 평균블록시간을 측정

100 : 500개의 블록을 생성한 후 계산된 평균블럭생성시간을 100회 반복하여 수행

결과값 출력 총 10000개의 블록을 생성하고 결과값으로 TPS(Transaction per second)값을 출력한다.

2.2.2블록 읽기 명령 설명

./dapp : 사고안전 및 사고안전 보상 블록체인 네트워크 접속프로그램

CcmCompRead : 블록읽기 체인코드를 실행하는 함수

0: 블록처리 시작번호

500: 블록처리 시간을 측정하기위해 한번에 처리하는 블록의 수로서 500개의 블록을 처리하고 평균블록 처리 시간을 측정

100 : 500개의 블록을 처리한 후 계산된 평균블럭 처리시간을 100회 반복하여 수행

결과값 출력: 총 10000개의 블록을 처리하고 결과값으로 TPS(Transaction per second)값을 출력한다.

2.2.3블록 변경 명령 설명

./dapp : 사고안전 및 사고안전 보상 블록체인 네트워크 접속프로그램

CcmCompUpdate : 블록변경 체인코드를 실행하는 함수

0: 블록처리 시작번호

500: 블록처리 시간을 측정하기위해 한번에 처리하는 블록의 수로서 500개의 블록을 처리하고 평균 블록 처리 시간을 측정

100 : 500개의 블록을 처리한 후 계산된 평균블럭 처리시간을 100회 반복하여 수행

결과값 출력: 총 10000개의 블록을 처리하고 결과값으로 TPS(Transaction per second)값을 출력한다.

2.2.4블록삭제 명령 설명

./dapp : 사고안전 및 사고안전 보상 블록체인 네트워크 접속프로그램

CcmCompDelete : 블록삭제 체인코드를 실행하는 함수

0: 블록처리 시작번호

500: 블록처리 시간을 측정하기 위해 한번에 처리하는 블록의 수로서 500개의 블록을 처리하고 평균 블록처리 시간을 측정

100 : 500개의 블록을 처리한 후 계산된 평균블록 처리시간을 100회 반복하여 수행

결과값 출력: 총 10000개의 블록을 처리하고 결과값으로 TPS(Transaction per second)값을 출력한다.

3 성능평가 결과

3.1 사고안전 네트워크 시험평가 결과

명령어	총수행시간 초	데이터 수	반복계산 수	평균값(초)	TPS
블록생성	84.4930	500	100	0.8449	591.7645
블록수정	59.9772	500	100	0.5997	833.6489
블록읽기	11.1360	500	100	0.1113	4489.9232
블록삭제	44.3137	500	100	0.4431	1128.3185

3.1.1 사고안전네트워크 블록생성 성능평가 결과 화면

```
qg@qg-System-Blockchain:~/fabric-samples/dapp$ ./dapp CcmDtgCreate 0 500 100
[./dapp CcmDtgCreate 0 500 100] CcmDtgCreate 0 500 100
[fabsdk/core] 2021/11/06 14:27:51 UTC - cryptosuite.GetDefault -> INFO No default cryptosuite
found, using default SW implementation
0 execution time: 8.574915ms
1 execution time: 17.473575ms
2 execution time: 34.362187ms
3 execution time: 14.177259ms
4 execution time: 15.95854ms
5 execution time: 2.176284417s
6 execution time: 967.23447ms
7 execution time: 753.813937ms
8 execution time: 993.970065ms
9 execution time: 1.045757195s
10 execution time: 707.24804ms
11 execution time: 1.065568206s
12 execution time: 769.384985ms
13 execution time: 991.109219ms
14 execution time: 1.035443392s
15 execution time: 777.319765ms
16 execution time: 1.000554847s
17 execution time: 753.456695ms
18 execution time: 979.771998ms
19 execution time: 1.000777947s
20 execution time: 855.006325ms
21 execution time: 839.240309ms
22 execution time: 909.04041ms
23 execution time: 1.048182137s
24 execution time: 799.997672ms
25 execution time: 987.516357ms
26 execution time: 979.319586ms
27 execution time: 835.949236ms
28 execution time: 849.662451ms
29 execution time: 778.351851ms
30 execution time: 948.637326ms
31 execution time: 887.267548ms
32 execution time: 867.536155ms
33 execution time: 776.337056ms
34 execution time: 980.92223ms
35 execution time: 816.146952ms
36 execution time: 980.819311ms
37 execution time: 858.124344ms
38 execution time: 793.991748ms
39 execution time: 1.031528548s
40 execution time: 769.957938ms
41 execution time: 962.732543ms
42 execution time: 829.771181ms
43 execution time: 651.072892ms
44 execution time: 857.72692ms
45 execution time: 723.720322ms
46 execution time: 921.779318ms
47 execution time: 791.323394ms
48 execution time: 811.159769ms
49 execution time: 882.830041ms
50 execution time: 880.185854ms
51 execution time: 899.975732ms
52 execution time: 820.67303ms
53 execution time: 861.243713ms
```

```

54 execution time: 937.283337ms
55 execution time: 881.665302ms
56 execution time: 863.532666ms
57 execution time: 739.778077ms
58 execution time: 991.785764ms
59 execution time: 873.586171ms
60 execution time: 877.293635ms
61 execution time: 751.596482ms
62 execution time: 789.991649ms
63 execution time: 925.28428ms
64 execution time: 794.488917ms
65 execution time: 905.209044ms
66 execution time: 906.291896ms
67 execution time: 934.21304ms
68 execution time: 837.287497ms
69 execution time: 874.122108ms
70 execution time: 827.509591ms
71 execution time: 934.482223ms
72 execution time: 940.435149ms
73 execution time: 892.553968ms
74 execution time: 901.462838ms
75 execution time: 717.309099ms
76 execution time: 953.197084ms
77 execution time: 903.24923ms
78 execution time: 740.963179ms
79 execution time: 885.4079ms
80 execution time: 783.221805ms
81 execution time: 908.833713ms
82 execution time: 783.648063ms
83 execution time: 906.605383ms
84 execution time: 862.168503ms
85 execution time: 887.864571ms
86 execution time: 854.555853ms
87 execution time: 914.952135ms
88 execution time: 871.372464ms
89 execution time: 896.840975ms
90 execution time: 911.190735ms
91 execution time: 733.229042ms
92 execution time: 1.014811622s
93 execution time: 747.223406ms
94 execution time: 981.677501ms
95 execution time: 758.564608ms
96 execution time: 917.41752ms
97 execution time: 897.305507ms
98 execution time: 826.17739ms
99 execution time: 859.45266ms
total Time : 84493065435ns Iterations : 100 average Time : 0.84493065435 sec Tps: 591.764540

```

3.1.2 사고안전네트워크 블록읽기 성능평가 결과 화면

```

qg@qg-System-Blockchain:~/fabric-samples/dapp$ ./dapp CcmDtgRead 0 500 100
0 execution time: 27.820724ms
1 execution time: 30.318008ms
2 execution time: 6.602562ms
3 execution time: 51.252465ms
4 execution time: 2.603409ms
5 execution time: 351.39957ms
6 execution time: 137.526399ms
7 execution time: 112.361222ms
8 execution time: 89.888317ms
9 execution time: 87.089939ms
10 execution time: 121.39948ms
11 execution time: 83.660249ms
12 execution time: 100.941341ms
13 execution time: 98.372829ms
14 execution time: 102.431317ms
15 execution time: 102.086973ms
16 execution time: 78.212243ms
17 execution time: 98.14768ms
18 execution time: 88.549823ms
19 execution time: 74.544778ms
20 execution time: 106.041034ms
21 execution time: 95.778004ms
22 execution time: 92.106242ms
23 execution time: 121.812074ms
24 execution time: 109.196824ms
25 execution time: 89.273746ms
26 execution time: 255.88313ms
27 execution time: 99.572668ms
28 execution time: 88.067565ms
29 execution time: 66.954763ms
30 execution time: 110.354043ms
31 execution time: 192.605882ms
32 execution time: 84.268213ms
33 execution time: 72.808892ms
34 execution time: 107.859824ms

```

```

35 execution time: 249.22497ms
36 execution time: 81.960908ms
37 execution time: 65.758881ms
38 execution time: 72.713572ms
39 execution time: 111.45343ms
40 execution time: 409.185956ms
41 execution time: 82.139269ms
42 execution time: 106.520217ms
43 execution time: 91.497664ms
44 execution time: 104.156723ms
45 execution time: 70.614232ms
46 execution time: 110.803758ms
47 execution time: 70.267912ms
48 execution time: 64.317636ms
49 execution time: 103.99729ms
50 execution time: 117.181007ms
51 execution time: 114.424831ms
52 execution time: 79.682237ms
53 execution time: 112.435604ms
54 execution time: 87.506213ms
55 execution time: 166.513875ms
56 execution time: 21.805254ms
57 execution time: 56.674004ms
58 execution time: 96.524815ms
59 execution time: 122.754964ms
60 execution time: 143.746452ms
61 execution time: 119.377407ms
62 execution time: 182.24857ms
63 execution time: 117.742383ms
64 execution time: 83.695684ms
65 execution time: 158.952323ms
66 execution time: 53.823646ms
67 execution time: 87.374917ms
68 execution time: 114.420087ms
69 execution time: 96.700642ms
70 execution time: 228.210863ms
71 execution time: 81.561957ms
72 execution time: 80.032301ms
73 execution time: 69.549763ms
74 execution time: 381.086855ms
75 execution time: 101.933204ms
76 execution time: 69.636777ms
77 execution time: 92.210314ms
78 execution time: 102.684664ms
79 execution time: 158.883132ms
80 execution time: 79.05565ms
81 execution time: 75.45119ms
82 execution time: 73.024862ms
83 execution time: 271.778667ms
84 execution time: 61.937299ms
85 execution time: 108.384943ms
86 execution time: 91.062316ms
87 execution time: 93.06803ms
88 execution time: 93.417897ms
89 execution time: 302.317758ms
90 execution time: 46.961803ms
91 execution time: 83.081963ms
92 execution time: 94.151784ms
93 execution time: 250.943857ms
94 execution time: 60.720938ms
95 execution time: 87.867938ms
96 execution time: 81.99889ms
97 execution time: 293.787102ms
98 execution time: 34.646663ms
99 execution time: 20.60875ms
total Time : 11136047695ns Iterations : 100 average Time : 0.11136047695 sec Tps: 4489.923299

```

3.1.3 사고안전네트워크 블록변경 성능평가 결과 화면

```

qg@qg-System-Blockchain:~/fabric-samples/dapp$ ./dapp CcmDtgUpdate 0 500 100
[./dapp CcmDtgUpdate 0 500 100] CcmDtgUpdate 0 500 100
[fabsdk/core] 2021/11/06 14:33:08 UTC - cryptosuite.GetDefault -> INFO No default cryptosuite
found, using default SW implementation
0실행시간: 36.395209ms
1실행시간: 3.341702ms
2실행시간: 60.802148ms
3실행시간: 1.022842ms
4실행시간: 1.155374ms
5실행시간: 2.054589164s
6실행시간: 837.856427ms
7실행시간: 841.439515ms
8실행시간: 677.479869ms

```

9	실행시간: 834.727811ms
10	실행시간: 670.008687ms
11	실행시간: 676.78099ms
12	실행시간: 574.205842ms
13	실행시간: 680.661377ms
14	실행시간: 701.95772ms
15	실행시간: 683.040382ms
16	실행시간: 657.443775ms
17	실행시간: 559.807034ms
18	실행시간: 692.034994ms
19	실행시간: 688.654913ms
20	실행시간: 653.806678ms
21	실행시간: 633.783442ms
22	실행시간: 570.834653ms
23	실행시간: 691.464548ms
24	실행시간: 691.041514ms
25	실행시간: 502.500093ms
26	실행시간: 601.073541ms
27	실행시간: 601.844253ms
28	실행시간: 669.262354ms
29	실행시간: 630.378498ms
30	실행시간: 491.281181ms
31	실행시간: 659.020272ms
32	실행시간: 560.238232ms
33	실행시간: 553.780804ms
34	실행시간: 623.867639ms
35	실행시간: 477.723585ms
36	실행시간: 673.008923ms
37	실행시간: 595.429301ms
38	실행시간: 529.085693ms
39	실행시간: 749.966903ms
40	실행시간: 480.59584ms
41	실행시간: 670.875649ms
42	실행시간: 553.600421ms
43	실행시간: 605.077924ms
44	실행시간: 727.073425ms
45	실행시간: 521.862266ms
46	실행시간: 678.661025ms
47	실행시간: 514.769252ms
48	실행시간: 677.150827ms
49	실행시간: 660.817338ms
50	실행시간: 480.109465ms
51	실행시간: 604.480518ms
52	실행시간: 663.184546ms
53	실행시간: 692.550381ms
54	실행시간: 685.043253ms
55	실행시간: 461.699263ms
56	실행시간: 617.303118ms
57	실행시간: 546.083819ms
58	실행시간: 589.105733ms
59	실행시간: 631.753782ms
60	실행시간: 487.808954ms
61	실행시간: 674.432174ms
62	실행시간: 603.657006ms
63	실행시간: 590.855332ms
64	실행시간: 539.628955ms
65	실행시간: 629.971044ms
66	실행시간: 604.585023ms
67	실행시간: 614.143525ms
68	실행시간: 593.419579ms
69	실행시간: 522.968418ms
70	실행시간: 649.637622ms
71	실행시간: 612.709938ms
72	실행시간: 610.624021ms
73	실행시간: 613.383063ms
74	실행시간: 467.870912ms
75	실행시간: 618.369744ms
76	실행시간: 571.533533ms
77	실행시간: 713.505921ms
78	실행시간: 597.541485ms
79	실행시간: 513.855209ms
80	실행시간: 618.810598ms
81	실행시간: 631.081845ms
82	실행시간: 591.657464ms
83	실행시간: 644.410758ms
84	실행시간: 685.843213ms
85	실행시간: 580.433367ms
86	실행시간: 624.211519ms
87	실행시간: 582.910282ms
88	실행시간: 652.262238ms
89	실행시간: 626.171868ms
90	실행시간: 503.676312ms
91	실행시간: 620.804407ms
92	실행시간: 586.457681ms
93	실행시간: 558.455077ms
94	실행시간: 542.995413ms
95	실행시간: 602.32235ms
96	실행시간: 602.225217ms
97	실행시간: 614.033435ms
98	실행시간: 567.371441ms
99	실행시간: 556.053633ms
total Time : 59977285303ns Iterations : 100 average Time : 0.59977285303 sec Tps: 833.648935	

3.1.4 사고안전네트워크 블록삭제 성능평가 결과 화면

```
qg@qg-System-Blockchain:~/fabric-samples/dapp$ ./dapp CcmDtgDelete 0 500 100
[./dapp CcmDtgDelete 0 500 100] CcmDtgDelete 0 500 100
[fabsdk/core] 2021/11/06 14:45:48 UTC - cryptosuite.GetDefault -> INFO No default cryptosuite
found, using default SW implementation
0실행시간: 17.396224ms
1실행시간: 10.060733ms
2실행시간: 46.393883ms
3실행시간: 11.834676ms
4실행시간: 1.618236ms
5실행시간: 1.19488651s
6실행시간: 545.473322ms
7실행시간: 665.43211ms
8실행시간: 534.487834ms
9실행시간: 702.559094ms
10실행시간: 513.144386ms
11실행시간: 429.088108ms
12실행시간: 547.348429ms
13실행시간: 467.23164ms
14실행시간: 538.89675ms
15실행시간: 429.450525ms
16실행시간: 497.567245ms
17실행시간: 537.869786ms
18실행시간: 412.432692ms
19실행시간: 546.295364ms
20실행시간: 429.472634ms
21실행시간: 429.310207ms
22실행시간: 591.795891ms
23실행시간: 449.428202ms
24실행시간: 502.173927ms
25실행시간: 375.596861ms
26실행시간: 406.060576ms
27실행시간: 530.46054ms
28실행시간: 391.005083ms
29실행시간: 478.644273ms
30실행시간: 381.98242ms
31실행시간: 367.317034ms
32실행시간: 477.60605ms
33실행시간: 455.45617ms
34실행시간: 417.213512ms
35실행시간: 478.551638ms
36실행시간: 390.07499ms
37실행시간: 495.042534ms
38실행시간: 423.628145ms
39실행시간: 396.078918ms
40실행시간: 435.35275ms
41실행시간: 482.571479ms
42실행시간: 448.986055ms
43실행시간: 461.303754ms
44실행시간: 469.176455ms
45실행시간: 468.652119ms
46실행시간: 411.63984ms
47실행시간: 406.103061ms
48실행시간: 497.532535ms
49실행시간: 383.565143ms
50실행시간: 461.875925ms
51실행시간: 491.717783ms
52실행시간: 361.176714ms
53실행시간: 505.739593ms
54실행시간: 379.895653ms
55실행시간: 521.832755ms
56실행시간: 403.411641ms
57실행시간: 408.344616ms
58실행시간: 437.767401ms
59실행시간: 354.389198ms
60실행시간: 456.26586ms
61실행시간: 356.684525ms
62실행시간: 536.905889ms
63실행시간: 379.533551ms
64실행시간: 500.56186ms
65실행시간: 479.843558ms
66실행시간: 355.906173ms
67실행시간: 537.965686ms
68실행시간: 383.255788ms
69실행시간: 368.031649ms
70실행시간: 478.114505ms
71실행시간: 491.716179ms
72실행시간: 433.306898ms
73실행시간: 544.494482ms
74실행시간: 408.014849ms
75실행시간: 469.812691ms
76실행시간: 372.9658ms
77실행시간: 487.784127ms
78실행시간: 443.363423ms
79실행시간: 538.638385ms
80실행시간: 474.117318ms
81실행시간: 374.276087ms
82실행시간: 477.270022ms
83실행시간: 386.251584ms
84실행시간: 504.796735ms
85실행시간: 415.124262ms
86실행시간: 520.653406ms
87실행시간: 445.490684ms
88실행시간: 392.882729ms
89실행시간: 426.819218ms
```

90실행시간:	508.072273ms
91실행시간:	392.864924ms
92실행시간:	515.478841ms
93실행시간:	490.572226ms
94실행시간:	495.083319ms
95실행시간:	377.554673ms
96실행시간:	382.388524ms
97실행시간:	542.144706ms
98실행시간:	397.284454ms
99실행시간:	464.033496ms
total Time :	44313726981ns Iterations : 100 average Time : 0.44313726981 sec Tps: 1128.318546

3.2 사고안전 보상 네트워크 시험평가 결과

명령어	총수행시간초	데이터 수	반복계산	평균값(초)	TPS
블록생성	72.7859	500	100	0.7278	686.9453
블록수정	55.4679	500	100	0.5546	901.4217
블록읽기	10.6938	500	100	0.1069	4675.5758
블록삭제	42.9578	500	100	0.4295	1163.9309

3.2.1 사고안전네트워크 블록생성 성능평가 결과 화면

qg@qg-System-Blockchain:~/fabric-samples/dapp\$./dapp CcmCompCreate 0 500 100	
[./dapp CcmCompCreate 0 500 100] CcmCompCreate 0 500 100	
[fabsdk/core] 2021/11/06 16:15:10 UTC - cryptosuite.GetDefault -> INFO No default cryptosuite found, using default SW implementation	
0	execution time: 5.454779ms
1	execution time: 41.014463ms
2	execution time: 2.802377ms
3	execution time: 22.73253ms
4	execution time: 3.01849ms
5	execution time: 1.845376678s
6	execution time: 886.38528ms
7	execution time: 731.462096ms
8	execution time: 896.962159ms
9	execution time: 721.660459ms
10	execution time: 811.413782ms
11	execution time: 788.370169ms
12	execution time: 662.158875ms
13	execution time: 831.68195ms
14	execution time: 649.936521ms
15	execution time: 861.951158ms
16	execution time: 784.340855ms
17	execution time: 715.579656ms
18	execution time: 781.499276ms
19	execution time: 782.755442ms
20	execution time: 812.392838ms
21	execution time: 741.661276ms
22	execution time: 664.598528ms
23	execution time: 902.393518ms
24	execution time: 702.624954ms
25	execution time: 851.146046ms
26	execution time: 656.373878ms
27	execution time: 746.822774ms
28	execution time: 897.70054ms
29	execution time: 718.948857ms
30	execution time: 812.658877ms
31	execution time: 657.981415ms
32	execution time: 776.381924ms
33	execution time: 876.933104ms
34	execution time: 677.349492ms
35	execution time: 895.953523ms
36	execution time: 651.287209ms
37	execution time: 793.19719ms
38	execution time: 809.66399ms
39	execution time: 719.659215ms
40	execution time: 842.112023ms
41	execution time: 658.156499ms
42	execution time: 873.547002ms
43	execution time: 810.358991ms
44	execution time: 652.942565ms
45	execution time: 859.332988ms
46	execution time: 660.956613ms
47	execution time: 850.22943ms
48	execution time: 821.759946ms
49	execution time: 587.685671ms
50	execution time: 724.934555ms
51	execution time: 635.168966ms
52	execution time: 847.262128ms
53	execution time: 668.231077ms
54	execution time: 767.934461ms
55	execution time: 729.474629ms
56	execution time: 824.584145ms
57	execution time: 660.464451ms
58	execution time: 770.516952ms
59	execution time: 829.473106ms
60	execution time: 621.188203ms
61	execution time: 869.402277ms
62	execution time: 726.209338ms
63	execution time: 769.649153ms
64	execution time: 714.722502ms

```

65 execution time: 693.960699ms
66 execution time: 902.32509ms
67 execution time: 666.005919ms
68 execution time: 812.473616ms
69 execution time: 803.873689ms
70 execution time: 628.15253ms
71 execution time: 789.015811ms
72 execution time: 731.449929ms
73 execution time: 885.954253ms
74 execution time: 685.269953ms
75 execution time: 515.222501ms
76 execution time: 812.801666ms
77 execution time: 539.634412ms
78 execution time: 825.395754ms
79 execution time: 659.414455ms
80 execution time: 782.056113ms
81 execution time: 848.279054ms
82 execution time: 511.165241ms
83 execution time: 791.570758ms
84 execution time: 819.273836ms
85 execution time: 766.361727ms
86 execution time: 764.449405ms
87 execution time: 639.485346ms
88 execution time: 821.922449ms
89 execution time: 642.2602ms
90 execution time: 838.572457ms
91 execution time: 676.428687ms
92 execution time: 628.256075ms
93 execution time: 726.230842ms
94 execution time: 792.845863ms
95 execution time: 794.755661ms
96 execution time: 769.718387ms
97 execution time: 792.338125ms
98 execution time: 648.144558ms
99 execution time: 812.345218ms
total Time : 72785994093ns  Iterations : 100  average Time : 0.72785994093 sec  Tps: 686.945347

```

3.2.2 사고안전네트워크 블록읽기 성능평가 결과 화면

```

qg@qg-System-Blockchain:~/fabric-samples/dapp$ ./dapp CcmCompRead 0 500 100
[./dapp CcmCompCreate 0 500 100] CcmCompRead 0 500 100
0 execution time: 13.287544ms
1 execution time: 36.54737ms
2 execution time: 16.289385ms
3 execution time: 8.77379ms
4 execution time: 9.173526ms
5 execution time: 166.71918ms
6 execution time: 130.774524ms
7 execution time: 124.984619ms
8 execution time: 114.496379ms
9 execution time: 108.482985ms
10 execution time: 152.755682ms
11 execution time: 110.055127ms
12 execution time: 71.89384ms
13 execution time: 134.084563ms
14 execution time: 185.961606ms
15 execution time: 121.047258ms
16 execution time: 93.828813ms
17 execution time: 103.434927ms
18 execution time: 83.106437ms
19 execution time: 85.926252ms
20 execution time: 100.188881ms
21 execution time: 112.684294ms
22 execution time: 72.321722ms
23 execution time: 106.50938ms
24 execution time: 75.560478ms
25 execution time: 100.507929ms
26 execution time: 74.958778ms
27 execution time: 112.629765ms
28 execution time: 95.632179ms
29 execution time: 88.003849ms
30 execution time: 201.257326ms
31 execution time: 19.105401ms
32 execution time: 27.049828ms
33 execution time: 111.270859ms
34 execution time: 135.909172ms
35 execution time: 119.083587ms
36 execution time: 223.471781ms
37 execution time: 95.445948ms
38 execution time: 281.924293ms
39 execution time: 540.873us
40 execution time: 185.700361ms
41 execution time: 87.437523ms
42 execution time: 91.164165ms
43 execution time: 108.335495ms
44 execution time: 69.881287ms
45 execution time: 155.600093ms
46 execution time: 71.813451ms
47 execution time: 94.922611ms
48 execution time: 93.833067ms
49 execution time: 109.725663ms
50 execution time: 48.07874ms
51 execution time: 99.66762ms

```

```

52 execution time: 62.317786ms
53 execution time: 130.13523ms
54 execution time: 141.795248ms
55 execution time: 68.094681ms
56 execution time: 117.714869ms
57 execution time: 107.22123ms
58 execution time: 291.139658ms
59 execution time: 84.789848ms
60 execution time: 89.985743ms
61 execution time: 81.997838ms
62 execution time: 78.951494ms
63 execution time: 292.283558ms
64 execution time: 107.17564ms
65 execution time: 65.408186ms
66 execution time: 79.587047ms
67 execution time: 167.832198ms
68 execution time: 112.8737ms
69 execution time: 73.208204ms
70 execution time: 109.521388ms
71 execution time: 71.280282ms
72 execution time: 169.726016ms
73 execution time: 197.333073ms
74 execution time: 76.370277ms
75 execution time: 77.718881ms
76 execution time: 97.832371ms
77 execution time: 150.92148ms
78 execution time: 96.049694ms
79 execution time: 83.295551ms
80 execution time: 90.520778ms
81 execution time: 170.591837ms
82 execution time: 80.401517ms
83 execution time: 82.162496ms
84 execution time: 71.273775ms
85 execution time: 118.797417ms
86 execution time: 234.38889ms
87 execution time: 124.097017ms
88 execution time: 51.06483ms
89 execution time: 86.900675ms
90 execution time: 82.302648ms
91 execution time: 190.242886ms
92 execution time: 85.021254ms
93 execution time: 91.889942ms
94 execution time: 94.851235ms
95 execution time: 197.076441ms
96 execution time: 124.795825ms
97 execution time: 67.632102ms
98 execution time: 77.353034ms
99 execution time: 48.103999ms
total Time : 10693869975ns Iterations : 100 average Time : 0.10693869975 sec Tps: 4675.575831

```

3.2.3 사고안전네트워크 블록변경 성능평가 결과 화면

```

qg@qg-System-Blockchain:~/fabric-samples/dapp$ ./dapp CcmCompUpdate 0 500 100
[./dapp CcmCompUpdate 0 500 100] CcmCompUpdate 0 500 100
[fabsdk/core] 2021/11/06 16:22:37 UTC - cryptosuite.GetDefault -> INFO No default cryptosuite
found, using default SW implementation
0실행시간: 19.790632ms
1실행시간: 24.805897ms
2실행시간: 2.268957ms
3실행시간: 48.774386ms
4실행시간: 12.337378ms
5실행시간: 2.166773881s
6실행시간: 792.190144ms
7실행시간: 705.795602ms
8실행시간: 802.379063ms
9실행시간: 658.772664ms
10실행시간: 652.041973ms
11실행시간: 670.362916ms
12실행시간: 528.059242ms
13실행시간: 693.57818ms
14실행시간: 566.063756ms
15실행시간: 620.63321ms
16실행시간: 540.605342ms
17실행시간: 510.104482ms
18실행시간: 584.516878ms
19실행시간: 412.537654ms
20실행시간: 590.023536ms
21실행시간: 580.402423ms
22실행시간: 478.424282ms
23실행시간: 617.905663ms
24실행시간: 498.223658ms
25실행시간: 573.326126ms
26실행시간: 575.597971ms
27실행시간: 464.515973ms
28실행시간: 583.327915ms
29실행시간: 480.403791ms
30실행시간: 622.59565ms
31실행시간: 580.507404ms
32실행시간: 486.586015ms
33실행시간: 626.260094ms
34실행시간: 462.102293ms

```

```

35실행시간: 645.710773ms
36실행시간: 458.128336ms
37실행시간: 546.711979ms
38실행시간: 569.670719ms
39실행시간: 551.337641ms
40실행시간: 541.877079ms
41실행시간: 490.477775ms
42실행시간: 585.524527ms
43실행시간: 591.218952ms
44실행시간: 623.945021ms
45실행시간: 457.345084ms
46실행시간: 602.736493ms
47실행시간: 651.949699ms
48실행시간: 581.677012ms
49실행시간: 489.797542ms
50실행시간: 553.793227ms
51실행시간: 553.730374ms
52실행시간: 631.000672ms
53실행시간: 512.727696ms
54실행시간: 579.276307ms
55실행시간: 532.484923ms
56실행시간: 608.004857ms
57실행시간: 538.295464ms
58실행시간: 615.787327ms
59실행시간: 722.84284ms
60실행시간: 491.648415ms
61실행시간: 633.149305ms
62실행시간: 589.056333ms
63실행시간: 655.222077ms
64실행시간: 567.024983ms
65실행시간: 459.552803ms
66실행시간: 641.044663ms
67실행시간: 499.536207ms
68실행시간: 512.747165ms
69실행시간: 623.523894ms
70실행시간: 464.407754ms
71실행시간: 612.645192ms
72실행시간: 444.884286ms
73실행시간: 661.770337ms
74실행시간: 485.096652ms
75실행시간: 570.172259ms
76실행시간: 577.100235ms
77실행시간: 586.024846ms
78실행시간: 548.864006ms
79실행시간: 445.737044ms
80실행시간: 561.773284ms
81실행시간: 597.07098ms
82실행시간: 509.200311ms
83실행시간: 610.088715ms
84실행시간: 463.281228ms
85실행시간: 645.139705ms
86실행시간: 549.051909ms
87실행시간: 442.668947ms
88실행시간: 596.036658ms
89실행시간: 412.764234ms
90실행시간: 618.377793ms
91실행시간: 619.594548ms
92실행시간: 471.133764ms
93실행시간: 562.818442ms
94실행시간: 497.670059ms
95실행시간: 578.475706ms
96실행시간: 548.165569ms
97실행시간: 617.004637ms
98실행시간: 505.720155ms
99실행시간: 524.040529ms
total Time : 55467932979ns Iterations : 100 average Time : 0.55467932979 sec Tps: 901.421728

```

3.2.4 사고안전네트워크 블록삭제 성능평가 결과 화면

```

qg@qg-System-Blockchain:~/fabric-samples/dapp$ ./dapp CcmDtgDelete 0 500 100
[./dapp CcmDtgDelete 0 500 100] CcmDtgDelete 0 500 100
[fabsdk/core] 2021/11/06 14:45:48 UTC - cryptosuite.GetDefault -> INFO No default cryptosuite
found, using default SW implementation
0실행시간: 17.396224ms
1실행시간: 10.060733ms
2실행시간: 46.393883ms
3실행시간: 11.834676ms
4실행시간: 1.618236ms
5실행시간: 1.19488651s
6실행시간: 545.473322ms
7실행시간: 665.43211ms
8실행시간: 534.487834ms
9실행시간: 702.559094ms
10실행시간: 513.144386ms
11실행시간: 429.088108ms
12실행시간: 547.348429ms
13실행시간: 467.23164ms
14실행시간: 538.89675ms
15실행시간: 429.450525ms
16실행시간: 497.567245ms
17실행시간: 537.869786ms

```

```

18실행시간: 412.432692ms
19실행시간: 546.295364ms
20실행시간: 429.472634ms
21실행시간: 429.310207ms
22실행시간: 591.795891ms
23실행시간: 449.428202ms
24실행시간: 502.173927ms
25실행시간: 375.596861ms
26실행시간: 406.060576ms
27실행시간: 530.46054ms
28실행시간: 391.005083ms
29실행시간: 478.644273ms
30실행시간: 381.98242ms
31실행시간: 367.317034ms
32실행시간: 477.60605ms
33실행시간: 455.45617ms
34실행시간: 417.213512ms
35실행시간: 478.551638ms
36실행시간: 390.07499ms
37실행시간: 495.042534ms
38실행시간: 423.628145ms
39실행시간: 396.078918ms
40실행시간: 435.35275ms
41실행시간: 482.571479ms
42실행시간: 448.986055ms
43실행시간: 461.303754ms
44실행시간: 469.176455ms
45실행시간: 468.652119ms
46실행시간: 411.63984ms
47실행시간: 406.103061ms
48실행시간: 497.532535ms
49실행시간: 383.565143ms
50실행시간: 461.875925ms
51실행시간: 491.717783ms
52실행시간: 361.176714ms
53실행시간: 505.739593ms
54실행시간: 379.895653ms
55실행시간: 521.832755ms
56실행시간: 403.411641ms
57실행시간: 408.344616ms
58실행시간: 437.767401ms
59실행시간: 354.389198ms
60실행시간: 456.26586ms
61실행시간: 356.684525ms
62실행시간: 536.905889ms
63실행시간: 379.533551ms
64실행시간: 500.56186ms
65실행시간: 479.843558ms
66실행시간: 355.906173ms
67실행시간: 537.965686ms
68실행시간: 383.255788ms
69실행시간: 368.031649ms
70실행시간: 478.114505ms
71실행시간: 491.716179ms
72실행시간: 433.306898ms
73실행시간: 544.494482ms
74실행시간: 408.014849ms
75실행시간: 469.812691ms
76실행시간: 372.9658ms
77실행시간: 487.784127ms
78실행시간: 443.363423ms
79실행시간: 538.638385ms
80실행시간: 474.117318ms
81실행시간: 374.276087ms
82실행시간: 477.270022ms
83실행시간: 386.251584ms
84실행시간: 504.796735ms
85실행시간: 415.124262ms
86실행시간: 520.653406ms
87실행시간: 445.490684ms
88실행시간: 392.882729ms
89실행시간: 426.819218ms
90실행시간: 508.072273ms
91실행시간: 392.864924ms
92실행시간: 515.478841ms
93실행시간: 490.572226ms
94실행시간: 495.083319ms
95실행시간: 377.554673ms
96실행시간: 382.388524ms
97실행시간: 542.144706ms
98실행시간: 397.284454ms
99실행시간: 464.033496ms
total Time : 44313726981ns Iterations : 100 average Time : 0.44313726981 sec Tps: 1128.318546

```

4. 성능평가 결과

4.1 사고안전 네트워크 시험평가 결과

평가대상	평가기준(TPS)	평가결과(TPS)	초과달성(TPS)	적부판정
블록생성	500	591.7	91.7	적합
블록수정	500	833.6	333.6	적합

블록읽기	500	4489.9	3989.9	적합
블록삭제	500	1128.3	628.3	적합

4.2 사고안전 네트워크 평가결과 그래프

4.2.1 사고안전 네트워크 평가결과 그래프 (선형 축척)



그림 95 사고안전 네트워크 평가결과 (선형 축척)

4.2.2 사고안전 네트워크 평가결과 그래프 (로그 축척)

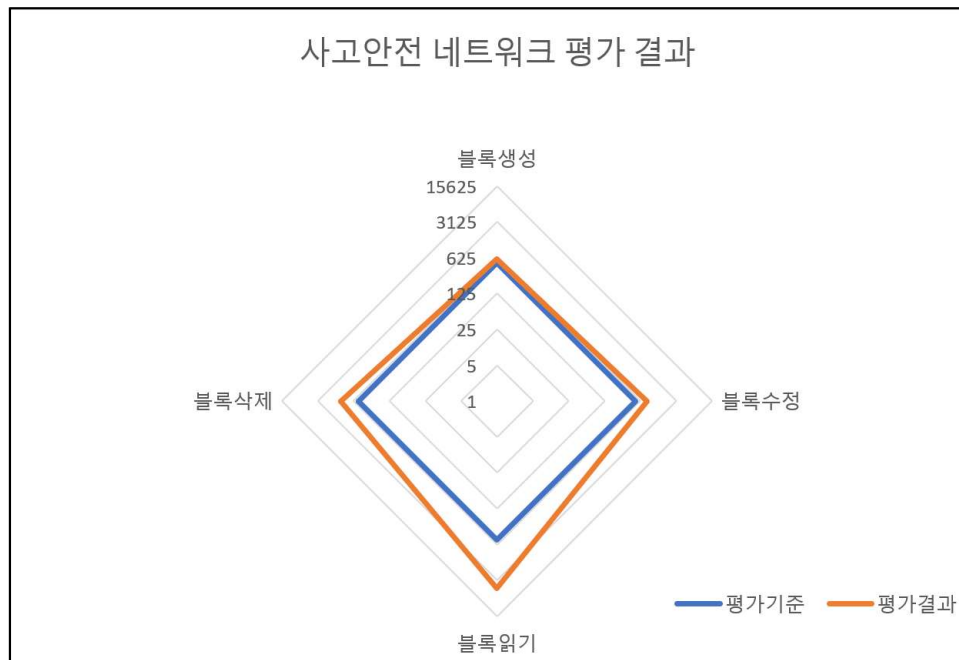


그림 96 사고안전 네트워크 (로그 축척)

4.3 사고안전 보상 네트워크 시험평가 결과

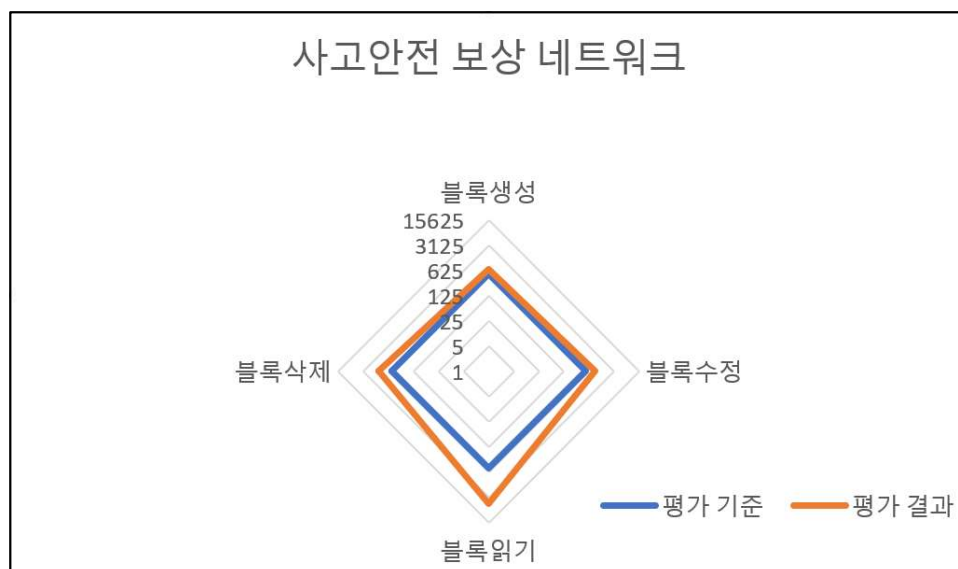
평가대상	평가기준(TPS)	평과결과(TPS)	초과달성(TPS)	적부판정
블록생성	500	686.9	186.9	적합
블록수정	500	901.4	401.4	적합
블록읽기	500	4675.5	4175.5	적합
블록삭제	500	1163.9	663.9	적합

4.4 사고안전 보상 네트워크 평가결과 그래프

4.4.1 사고안전 보상 네트워크 평가결과 그래프 (선형 축척)



4.4.1 사고안전 보상 네트워크 평가결과 그래프 (로그 축척)



3. 연구개발과제의 수행 결과 및 목표 달성 정도

1) 연구수행 결과

(1) 정성적 연구개발성과

○ 블록체인 적용

- 라이트 블록체인 개발적용으로 온 오프라인 위변조 방지 시스템 구축
- 하이퍼레저패브릭 블록체인의 단점이 블록처리성능을 병렬처리 기법을 활용하여 처리속도를 향상

○ 디지털운행기록장치 + ADAS + 블록체인 기반 안전운행 보상 All in One 시스템

- 기존 차량운행장치의 운행기록 데이터는 변조 가능 > 본 과제를 통해 데이터 위변조 불가능
- 안전운행 지원시스템(전방추돌방지시스템, 차선이탈경보시스템)이 하나의 장치로 포함되어 개발
- 화물차 운전자의 교통안전운행 행위에 대한 보상을 통해 대형 교통사고 감소에 기여

(2) 정량적 연구개발성과

구분		1차년도(2019년)	2차년도(2020년)	3차년도(2021년)	계
국외논문 게재	SCI	-	1건	-	1건
	비SCI	-	-	-	-
	계	-	1건	-	1건
국내논문 게재	SCI	-	-	1건	1건
	비SCI	-	-	-	-
	계	-	-	1건	1건
특허출원	국내	2건	1건	-	3건
	국외	-	-	-	-
	계	-	-	-	-
특허등록	국내	-	-	3건	3건
	국외	-	-	-	-
	계	-	-	-	-
기술이전	계약건수	-	-	-	-
	계약액	-	-	-	-
	징수액	-	-	-	-
국제 표준화	기고서 제출	-	-	-	-
	기고서 채택	-	-	-	-
	표준안 채택	-	-	-	-
국내 표준화	기고서 제출	-	-	-	-
	기고서 채택	-	-	-	-
	표준안 채택	-	-	-	-
등록·기탁 대상 성과물	보고서 원문	데이터 수집 보고서 1건	데이터 수집 보고서 1건	-	2건
	연구시설·장비	-	-	-	-
	기술요약정보	-	-	-	-
	생명 자 원	생명정보	-	-	-
		생물자원	-	-	-
		신품종	-	-	-
	소프트웨어	S/W 5건	S/W 6건	S/W 12건	23건
	화학물	-	-	-	-
기타	시제품	시제품 2건	시제품 2건	시제품 3건	7건
	성과 홍보	-	-	-	-
	일자리 창출	-	-	-	-

(3) 세부 정량적 연구개발성과

[과학적 성과]

☐ 논문(국내외 전문 학술지) 게재

번호	논문명	학술지명	주저자명	호	국명	발행기관	SCIE 여부 (SCIE/비SCIE)	게재일	등록번호 (ISSN)	기여율
1	A Tamper-Resistant Algorithm Using Blockchain for the Digital Tachograph	Electronics	김용배 백주용 김종원	Volume 10 Issue 5, 581	Switzerland	MDPI	SCI	2021. 3월	2079-9292	100%
2	하이퍼레저 패브릭을 이용한 화물차 디지털 운행기록 단말기의 안전운행 보상시스템 구현	인터넷정보 학회논문지	김용배 백주용 김종원		대한민국	한국인터넷 정보학회	비SCIE (KCI 등재지)	2022.02.05 (투고 완료) 2022. 2월말 발간 예정	1598-0170	100%

☐ 국내 및 국제 학술회의 발표

번호	회의 명칭	발표 내용	발표자	발표 일시	장소	국명

☐ 기술 요약 정보

번호	연도	기술명	요약 내용	기술 완성도	등록 번호	활용 여부	미활용사유	연구개발기관 외 활용 여부	허용방식

☐ 보고서 원문

번호	연도	보고서 명	발간일	등록 번호
1	2019	화물차 운전자(차량)의 안전운행 행위에 대한 보상을 위한, 블록체인 기반 디지털운행기록단말 장치와 안전운행 보상 시스템 개발 1차년도 실적보고(2019)	2019. 12. 05	
2	2020	화물차 운전자(차량)의 안전운행 행위에 대한 보상을 위한, 블록체인 기반 디지털운행기록단말 장치와 안전운행 보상 시스템 개발 2차년도 실적보고(2020)	2020. 12. 10	

☐ 생명자원(생물자원, 생명정보)/화합물

번호	생명자원(생물자원, 생명정보)/화합물 명	등록/기탁 번호	등록/기탁 기관	발생 연도

[기술적 성과]

☐ 지식재산권(특허, 실용신안, 의장, 디자인, 상표, 규격, 신품종, 프로그램)

번호	구분	지식재산권 등 명칭 (건별 각각 기재)	국명	출원			등록			기여 율	활용 여부
				출원인	출원일	출원번호	등록인	등록일	등록번호		
1	특허	블록체인 기반의 운행기록 분석과 안전운전 판단을 실행하는 DTG 단말기 및 방법	대한 민국	(주)퀀텀 게이트	2019 09.06	10-2019-01 10840	(주)퀀텀 게이트	2020 03.03	제10-2086966호	100%	활용 예정
2	특허	블록체인 기반의 안전운전 에 대한 보상을 제공하는 시스템 및 방법	대한 민국	(주)퀀텀 게이트	2019 09.06	10-2019-01 10851	(주)퀀텀 게이트	2020 03.05	제10-2226951호	100%	활용 예정
3	특허	블록체인 기반의 차량 운 행 정보 관리 방법 및 시 스템	대한 민국	(주)퀀텀 게이트	2020 06.15	10-2020-00 72565				100%	활용 예정
4	특허	블록체인 네트워크에 기반 한 운행기록 데이터의 관 리 방법, 이를 수행하기 위한 장치 및 시스템	대한 민국	(주)퀀텀 게이트	2021 07.09	10-2021-00 90565				100%	활용 예정

○ 지식재산권 활용 유형

번호	제품화	방어	전용실시	통상실시	무상실시	매매/양도	상호실시	담보대출	투자	활용 예정	기타
										✓	

□ 저작권(소프트웨어, 서적 등)

번호	저작권명	창작일	저작자명	등록일	등록번호	저작권자명	기여율
1	DTG와 차량 CAN 통신프로그램	2019.11.20	주식회사 퀀텀게이트	2019.12.04	C-2019-037438	주식회사 퀀텀게이트	100%
2	J1939/J1708 프로 그램	2019.11.20	주식회사 퀀텀게이트	2019.12.04	C-2019-037439	주식회사 퀀텀게이트	100%
3	DTG주변장치(GPS, 가속도센서, 자이 로센서 등등)컨트 롤 프로그램	2019.11.20	주식회사 퀀텀게이트	2019.12.04	C-2019-037437	주식회사 퀀텀게이트	100%
4	디지털 운행기록 계 데이터 위 변 조 바지를 위한 라이트 블록체인 프로그램	2019.11.11	주식회사 퀀텀게이트	2019.11.28	C-2019-036314	주식회사 퀀텀게이트	100%
5	디지털 운행기록 계의 일일 데이터 셋	2019.11.11	주식회사 퀀텀게이트	2019.11.28	D-2019-000071	주식회사 퀀텀게이트	100%
6	급차로 변경 확인 기능 프로그램	2020.10.20	주식회사 퀀텀게이트	2020.10.22	C-2020-037150	주식회사 퀀텀게이트	100%
7	안전거리 미확보 확인기능 프로그 램	2020.10.20	주식회사 퀀텀게이트	2020.10.22	C-2020-037152	주식회사 퀀텀게이트	100%
8	전방차량 추돌방 지 프로그램	2020.10.20	주식회사 퀀텀게이트	2020.10.22	C-2020-037049	주식회사 퀀텀게이트	100%
9	차선이탈 경보 기 능 프로그램	2020.10.20	주식회사 퀀텀게이트	2020.10.22	C-2020-037151	주식회사 퀀텀게이트	100%
10	카메라 영상 저장 프로그램	2020.10.20	주식회사 퀀텀게이트	2020.10.22	C-2020-037148	주식회사 퀀텀게이트	100%
11	안전운전 블록체 인 네트워크의 스 마트 컨트랙트(체 인코드)	2020.10.20	주식회사 퀀텀게이트	2020.10.20	C-2020-038443	주식회사 퀀텀게이트	100%

☐ 신기술 지정

번호	명칭	고시일	보호기간	지정번호

☐ 기술 및 제품 인증

번호	인증 분야	인증기관	인증 내용		인증 획득일	국가명
			인증명	인증번호		

☐ 표준화

○ 국내표준

번호	인증구분	인증여부	표준명	표준인증기구명	제안주체	표준종류	제안/인증일자

○ 국제표준

번호	표준화단계 구분	표준명	표준기구명	표준분과명	의장단 활동여부	표준특허 추진여부	표준개발 방식	제안자	표준화 번호	제안일자

[경제적 성과]

☐ 시제품 제작

번호	시제품명	출시/제작일	제작업체명	설치장소	이용분야	사업화 소요 기간	인증기관 (해당 시)	인증일 (해당 시)

☐ 현장적용

번호	적용기술명	적용일	발주기관명	적용 연구개발기관명	적용현장명(장소)

□ 기술 실시(이전)

번호	기술 이전 유형	기술 실시 계약명	기술 실시 대상 기관	기술 실시 발생일	기술료	
					발생 연도	발생액(천원)

□ 사업화 투자실적

번호	연도	투자 자금 성격*	추가 연구개발 투자(천원)	설비 투자(천원)	기타 투자(천원)	합계(천원)

□ 사업화 현황

번호	사업화 방식	사업화 형태	지역	사업화명	내용	업체명	매출액		매출 발생 연도	기술 수명
							국내(천원)	국외(달러)		

□ 매출 실적(누적)

번호	사업화명	발생 연도	매출액		합계	산정 방법
			국내(천원)	국외(달러)		
합계						

□ 사업화 계획 및 무역 수지 개선 효과

성과			블록체인 기반 안전운행보상 디지털운행기록장치 UXDTG(제품명)		
사업화 계획	사업화 소요기간(년)		2~3년		
	소요예산(천원)		800,000		
	예상 매출규모(천원)		현재까지	3년 후	5년 후
			-	100,000	1,000,000
	시장 점유율	단위(%)	현재까지	3년 후	5년 후
		국내	-	0.1%	3~5%
		국외			0.01%
향후 관련기술, 제품을 응용한 타 모델, 제품 개발계획		1. 단말기 개발(보급형, 스마트형) 2. 관제SW 개발 3. 기업용 안전운행보상시스템 개발			
무역 수지 개선 효과(천원)	수입대체(내수)	현재	3년 후	5년 후	
		-	-	100,000	
	수 출	-	-	50,000 (테스트 마켓 진출)	

□ 고용 창출

번호	사업화명	사업화 업체명	고용창출 인원(명)		합계(명)
			yyyy년	yyyy년	
합계					

□ 고용 효과

구분			고용 효과(명)
고용 효과	개발 전	연구인력	
		생산인력	
	개발 후	연구인력	
		생산인력	

□ 비용 절감(누적)

번호	사업화명	발생 연도	산정방법	비용 절감액(천원)
합계				

□ 경제적 파급 효과

(단위 : 천원/년)

구분	사업화명	수입 대체	수출 증대	매출 증대	생산성 향상	고용 창출 (인력 양성 수, 명)	기타
해당 연도							
기대 목표							

□ 산업 지원(기술지도)

번호	내용	기간	참석 대상	장소	인원(명)

□ 기술 무역

(단위 : 천원)

번호	계약 연월	계약 기술명	계약 업체명	계약업체 국가	총 계약액	기 징수액	해당 연도	징수액	향후 예정액	수출/ 수입

[사회적 성과]

□ 법령 반영

번호	구분 (법률/시행령)	활용 구분 (제정/개정)	명 칭	해당 조항	시행일	관리 부처	제정/개정 내용

☐ 정책활용 내용

번호	구분 (제안/채택)	정책명	관련 기관 (담당 부서)	활용 연도	채택 내용

☐ 설계 기준/설명서(시방서)/지침/안내서에 반영

번호	구분 (설계 기준/설명서/지침/안내서)	활용 구분 (신규/개선)	설계 기준/설명서/ 지침/안내서 명칭	반영일	반영 내용

☐ 신진연구자 채용

번호	채용 연구개발기관명	채용일	근로기간	채용인원(명)	채용인원 중 청년인력(명)

☐ 전문 연구 인력 양성

번호	분류	연도	현황(명)										
			학위별				성별		지역별				
			박사	석사	학사	기타	남	여	수도권	충청권	영남권	호남권	기타

☐ 산업 기술 인력 양성

번호	연도	프로그램명	프로그램 내용	교육 기관	교육 개최 횟수	총 교육 시간	총 교육 인원

☐ 다른 국가연구개발사업에의 활용

번호	연도	중앙행정기관명	사업명	연구개발과제명	연구책임자	연구개발비(천원)

☐ 국제화 협력성과

번호	구분 (유치/파견)	기간	국가	학위	전공	내용

☐ 홍보 실적

번호	홍보 유형	매체명	제목	홍보일

☐ 포상 및 수상 실적

번호	종류	포상명	포상 내용	포상 대상	포상일	포상 기관

[인프라 성과]

□ 연구시설·장비

번호	구축 기관	연구시설/ 연구장비명	규격 (모델명)	개발여부 (○/×)	연구시설·장비 종합정보시스템* 등록여부	연구시설·장비 종합정보시스템* 등록번호	구축일자 (YY.MM.DD)	구축비용 (천원)	비고 (설치 장소)

[그 밖의 성과](해당 시 작성)

○

-

○

-

(4) 계획하지 않은 성과 및 관련 분야 기여사항(해당 시 작성)

○

-

○

-

<참고 1> 연구개발성과 실적 증빙자료 예시

성과유형	첨부자료 예시
연구논문	논문 사본(저자, 초록, 사사표기를 확인할 수 있는 부분 포함, 연구개발과제별 중복 첨부 불가)
지식재산권	지식재산권 등록증(또는 출원서) 사본(발명인, 발명의 명칭, 연구개발과제 출처 포함)
제품개발(시제품)	제품개발사진 등 시제품 개발 관련 증빙자료
기술이전	기술이전 계약서, 기술실시 계약서, 기술료 입금 내역서 등
사업화 (상품출시, 공정개발)	사업화된 제품사진, 매출액 증빙서류(세금계산서, 납품계약서 등 매출 확인가능 내부 회계자료) 등

<참고 2> 「국가연구개발혁신법」 시행령 제33조 제4항 및 별표 4에 따른 연구개발성과의 등록·기탁 대상과 범위

구분	대상	등록 및 기탁 범위
등록	논문	국내외 학술단체에서 발간하는 학술(대회)지에 수록된 학술 논문(전자원문 포함)
	특허	국내외에 출원 또는 등록된 특허정보
	보고서원문	연구개발 연차보고서, 단계보고서 및 최종보고서의 원문
	연구시설 · 장비	국가연구개발사업을 통하여 취득한 3천만 원 이상(부가가치세, 부대비용 포함) 연구시설·장비 또는 공동 활용이 가능한 모든 연구시설·장비
	기술요약정보	연차보고, 단계보고 및 최종보고가 완료된 연구개발성과의 기술을 요약한 정보
	생명자원 중 생명정보	서열·발현정보 등 유전체정보, 서열·구조·상호작용 등 단백질체정보, 유전자(DNA)칩·단백질칩 등 발현체 정보 및 그 밖의 생명정보
	소프트웨어	창작된 소프트웨어 및 등록에 필요한 관련 정보
기탁	표준	「국가표준기본법」 제3조에 따른 국가표준, 국제표준으로 채택된 공식 표준정보(소관 기술위원회를 포함한 공식 국제표준화기구(ISO, IEC, ITU)가 공인한 단체 또는 사실표준화기구에서 채택한 표준정보를 포함)
	생명자원 중 생물자원	세균, 곰팡이, 바이러스 등 미생물자원, 인간 또는 동물의 세포·수정란 등 동물자원, 식물세포·종자 등 식물자원, DNA, RNA, 플라스미드 등 유전체자원 및 그 밖의 생물자원
	화합물	합성 또는 천연물에서 추출한 유기화합물 및 관련 정보
	신품종	생물자원 중 국내외에 출원 또는 등록된 농업용 신품종 및 관련 정보

2) 목표 달성 수준

연구개발목표	달성내용	달성도(%)
○ 디지털 운행 기록 장치 (기본 Spec.)+ADAS(전방추돌, 차선이탈)+블록체인 기반안전운행 보상 시스템이 하나로 결합된 All in One 제품개발 ○ DTG 단말용 블록체인 네트워크 및 블록체인기반의 안전운전 보상 시스템 개발	○ HD카메라 탑재 ADAS(전방추돌, 차선이탈) 개발로 안전운행을 지원하고 영상기능을 활용하여 안전운행 이행 상황 검증용으로 개발(차선, 안전거리) ○ SDCN(Safe Driving Channel Network) 시스템의 사고 및 안전운전 블록체인 네트워크 개발 ○ SDCN 보상 블록체인 네트워크 개발 ○ DTG 단말기 데이터 위·변조 방지를 위한 라이트 블록체인 개발 ○ 유사환경 및 제한된 실제 환경에서 시스템의 측정 및 신뢰도 검증	100%

4. 목표 미달 시 원인분석(해당 시 작성)

1) 목표 미달 원인(사유) 자체분석 내용

○

-

○

-

2) 자체 보완활동

○

-

○

-

3) 연구개발 과정의 성실성

○

-

○

-

5. 연구개발성과의 관련 분야에 대한 기여 정도

○ 본 연구에서 개발된 경량 블록체인 시스템은 온, 오프라인 모두에서 동작할 수 있으며, 모바일 기기에 적합한 특징을 가지고 있다.

- 차량운행 중 네트워크 접속이 단절된 경우 서버와 단말기 간의 상호인증과정을 수행할 수 없게 된다. 이 경우 서버에서 제공한 랜덤시드(Random Seed)를 활용하여 오프라인에서도 안전하게 사용할 수 있는 알고리즘을 개발 적용하였다.

(본 연구 성과물인 'Electronics' 발표 논문 내용 인용 / 상세 내용, 첨부 논문 전문 참고)

- 차량의 운행기록 장치는 자동차의 운행상황을 속도, 거리, 브레이크 작동 상황, 가속도, GPS정보 등을 1초 간격으로 기록하고 있으며, 사고 상황 즉 충돌 데이터는 1/100초 주기로 사고전후 10초 이상 가속도, 차량 진행방향 등의 정보를 기록하도록 하고 있다. 운행기록장치를 장착한 차량은 주기적으로 운행 데이터를 관리기관에 업로드 하여야 하는 의무가 주어지는데, 이때 usb와 같은 보조 저장장치에 복사하거나 온라인 무선통신을 이용하여 데이터를 업로드 하게 된다. 운행 기록된 내용에 문제가 있는 경우 업로드 과정에서 데이터의 위·변조에 노출되는 위험이 있다.

특히 모바일 기기의 성능향상을 바탕으로 실시간 위치전송이 가능한 디지털 운행기록장치가 개발되면서 실시간 위치추적 서비스와 같은 부가서비스가 제공되고 있어 물류회사의 배송정보 확인 등에 활용되고 있다. 이러한 환경의 변화는 서비스의 형태만 아니라 하드웨어에도 적용되기 시작하고 있다. 지금까지

는 인증된 전용장비를 활용한 서비스가 주류를 이루고 있었으나, 모바일기기에 앱을 다운로드하여 실행하는 형태의 소프트웨어 기반의 서비스도 개발되고 있다.

본 연구에서는 온·오프라인의 환경에서 디지털 운행기록장치의 데이터의 위·변조를 방지할 수 있는 블록체인기반의 데이터 위·변조 방지 기술을 제안하였다. 제안하는 알고리즘은 디지털 운행기록장치의 저 수준 하드웨어에서 동작할 수 있도록 light blockchain과 온·오프라인 상황에서도 데이터의 위·변조를 방지할 수 있는 새로운 데이터 기록 메커니즘을 제안하였다. 제안된 light blockchain은 인코딩 시간이 평균적으로 1.85msec/MB, 디코딩 시간이 1.65msec/MB가 소요되었고 통계검정을 통한 outlier를 제거하면 인코딩 시간 평균이 1.32msec/MB, 디코딩 시간 평균이 1.29msec/MB로 추정되었으며, 위·변조 검증 실험에서는 모든 위·변조된 데이터를 검출할 수 있었다.

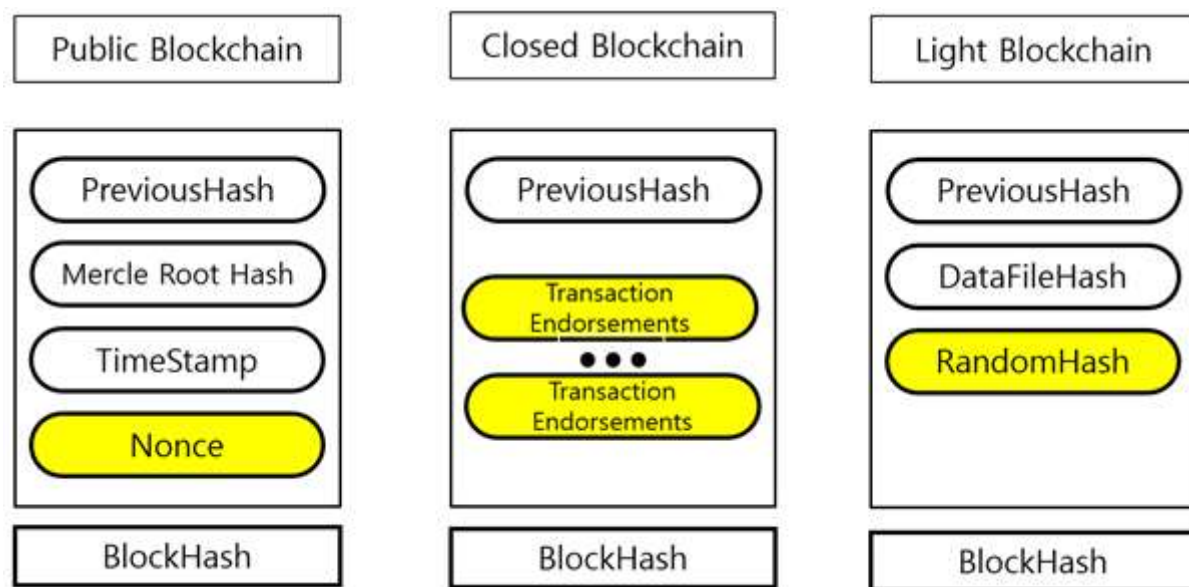


Figure. Light Blockchain Architecture

○ ‘블록체인 가속 알고리즘’ 개발로 처리속도 향상

블록읽기 성능은 연구초기 12.51TPS에서 1,040TPS로 **83배 이상 향상**,

블록 삭제의 경우 연구초기 9.21TPS에서 1,863.97TPS로 **200배 이상의 성능향상**.

(본 연구 성과물인 ‘인터넷정보학회논문지’ 발표 논문 내용 인용 / 상세 내용, 첨부 논문 전문 참고)

- 블록체인은 여러 가지 장점에도 불구하고 느린 처리속도가 걸림돌이다. 비트코인의 경우 하나의 새로운 블록을 구성하려면 약 10분, Litecoin 5분, Ethereum 15초 정도 소요되며, 거래속도는 비트코인이 7TPS, Litecoin 56TPS, Ethereum 20TPS, 정도이다.
- 본 연구에서 개발된 경량 블록체인 시스템은 온, 오프라인 모두에서 동작할 수 있는 특징을 가지고 있다. 차량 운행 중 네트워크 접속이 단절된 경우 서버와 단말기 간의 상호인증과정을 수행할 수 없게 된다. 이 경우 서버에서 제공한 랜덤시드(Random Seed)를 활용하여 오프라인에서도 안전하게 사용할 수 있는 알고리즘을 개발 적용하였다.
- 본 연구에서도 화물차를 대상으로 한 블록체인 시스템으로 수만 대의 차량을 처리하는데 10TPS 내외의 속도로는 실제 활용이 불가능하다. 속도향상을 위해서 Go언어의 병렬처리기법을 이용하여 2,500개의 프로세스를 유지하도록 제어하였다. 본 연구에서는 블록의 읽기, 쓰기, 변경, 삭제까지 블록체인 응용에 사용되는 최소한의 기능에 대하여 성능을 평가하였으며, 그 결과는 표 14와 같다.

(표 91) 순차 대 병렬처리 성능 비교

(Table 14) Sequential vs Concurrent Transaction Performance

명령어	가 속 미적용	가속 적 용	속도 비교 (배)
안전채널 블록생성	12.51	1040.06	83.14
안전채널 블록읽기	91.99	2496.10	27.13
안전채널 블록수정	13.50	2067.15	153.12
안전채널 블록삭제	13.58	2206.22	162.46
보상채널 블록생성	12.51	955.27	76.36
보상채널 블록읽기	91.87	2515.27	27.38
보상채널 블록수정	13.56	1792.08	132.16
보상채널 블록삭제	9.21	1863.97	202.39

- 블록읽기 성능은 12.51TPS에서 1,040TPS로 83배 이상 향상되었으며, 블록 삭제의 경우 9.21TPS에서 1,863.97TPS로 200배 이상의 성능향상을 나타내고 있어 실용화할 수 있게 하였다.

6. 연구개발성과 관리 및 활용 계획

○ 활용분야 및 파급효과

1. 기술적 측면

- ① 고속도로 사고다발지역에서 상용차 운행행태 데이터 분석, 정책추진 시 근거자료 활용
- ② 연구성과인 라이트 블록체인 기술활용, 디지털운행기록장치(DTG) 위반조방지 및 기능고도화
- ③ 교통안전공단 디지털운행기록장치(DTG) 효율적 관리
- ④ 차량운행기록장치+ADAS(전방추돌, 차선이탈)+블록체인 기반 안전운행 보상시스템이 하나로 결합된 All in One 제품으로 신규시장 창출 가능

2. 경제적·산업적 측면

- ⑤ 대형 상용차(화물차, 특수차량, 장거리 운행 버스 등)의 안전운행 지원 대책
- ⑥ 향후 관련법규, 기술적인문제, 보상체계 등이 해결되면 하이패스 단말기 대체 가능
- ⑦ 보상으로 지급받은 전자바우처의 생태계 활성화로 블록체인 기반의 핀테크 경제 활성화에 기여(휴게소, 과태료, 통행료, 차량정비 등에서 보상으로 지급받은 전자바우처 활용)
- ⑧ 국내 신규시장 창출을 통한 수출 기대

3. 사회적 측면

- ⑨ 화물차 및 버스의 대형 교통사고 감소로 국민의 안전과 생명보호
- ⑩ 열악한 근무환경과 저임금에 시달리는 대형 상용차 운전자들에게 안전운전 동기를 부여하고 안전운전 습관 개선에 기여

○ 활용 방안

1. 사업화를 통한 주요 예상 고객

- ⑪ 전국화물운송조합, 버스운송조합
- ⑫ 대형 운송법인 및 유통회사(CJ 대한통운, SPC그룹, 이마트 등)
- ⑬ 특수화물 차량(시멘트, 탱크로리, 컨테이너, 정유차량, 의약품, 농수산물, 화학약품 등)
- ⑭ 택시, 공유 카 플랫폼 서비스 업체 등
- ⑮ 공공시장 진출(혁신제품 등록 목표)

2. 사업화 모형(BM)을 통한 수익 창출 방안

- ⑯ 단말기 제품경쟁력 : 디지털운행기록장치+ADAS(전방추돌, 차선이탈)+안전운행 보상시스템
 - ⑰ 각 사업자별 특화된 차량관제 시스템 제공으로 시스템 판매, 교육, 유지보수 수익
 - ⑱ 블록체인 안전운행보상 스마트컨트랙트 시스템운영 관리 수익
 - ⑲ 전자바우처 생태계 조성을 통한 전자바우처 거래 수수료
-

1. 공공/민간 사업화 모델에 따라 보상체계(정책)에 대한 논의가 다를 것으로 예상되며, 이에 대한 한계 사항 제시

1.1. 공공부문 사업화 모델에 따른 보상체계

가. 제도 운영 당위성

- 화물차 운전자의 장시간 노동 투입 대비 열악한 수입으로 인한 과적, 과속 등 무리한 운행 방지
- 단속 위주의 정책에서 벗어나, 운전자 스스로 안전 운행에 참여하도록 동기부여
- 안전운행 지원사업에 소요되는 각종 비용(장비 지원 등)이 정확하게 어떻게 사용되었는지를 추적하기가 현실적으로 어렵고 효과측정도 불가
- 모호하게 운영 지원되는 여러 가지 국가지원정책 대비, 본 사업의 투입비용과 효과측정이 정확한 데이터 기반으로 확인 가능

나. 제원마련 방안

- 행안부, 국토부 등에서 화물차 안전운행에 관한 여러 가지 지원책 활용
- 운행기록장치 의무장착(200억 이상), ADAS(300억 이상) 등 안전운행에 필요한 장비 지원사업
- 심야 통행료 할인(연간 70억 이상), 연비왕선발 포상, 안전운행 관련 포상 등
- 상기 국가에서 지원하는 여러 가지 지원비용의 일부 예산 활용 가능

다. 보상체계 안

- 전국 고속화 도로 사고다발지역을 우선하여 안전운행 구간 및 보상구간 설정
- 위험도에 따라 보상 수준 차별화
- 출발지 ~ 도착지까지 지속적인 안전운행 유도를 위한 체계적인 구간 설정
- 운행기록 데이터를 근거로 보상이 주어지고, 보상받은 전자바우처를 어디에 어떻게 사용했는지 추적 가능하기 때문에 투명한 제도운영 가능

라. 제도 운영을 위한 예상 문제점

- 안전운행 보상지급을 위한 지속적인 예산확보
- 일반 운전자와 형편성(이미 화물차의 여러 가지 지원정책실시, 큰 이슈는 아닌 것으로 판단)
- 정확한 소요 예산 도출, 운영을 위한 기관설립(교통안전공단 내 금융기관 성격의 부처 신설)
- 운전자와 제도 운영기관 간의 보상내용에 대한 다툼 예상, 이에 따른 소모적인 행정력 발생 가능

1.2. 민간부문 사업화 모델에 따른 보상체계

가. 제도 운영 필요성

- 소속 운전자의 안전운행으로 인한 보상투입대비 비용 절감(대인 대물 사고처리, 보험료 인상 등)
- 위반조 되지 않은 운행 기록데이터 분석(사고원인, 운전자 행태 분석)
- 소속 운전자의 소속감, 조직에 대한 만족도 향상을 통한 근무 효율증가

나. 제원마련 방안

- 운수 법인의 통상적인 사고처리비용, 보험료 등 안전운행 관리 투입비용의 일부 재원활용
- 일정기간 시범운행을 통한 효과 평가 후 확대(보상 제도실시 전후, 비용 절감 및 만족도 평가)

다. 보상 체계 안

- 전국 고속화 도로 사고다발지역을 우선하여 안전운행 구간 및 보상구간 설정
- 위험도에 따라 보상 수준 차별화
- 소속 운전자간의 안전운행 실적 경쟁(우수 성과자 추가적인 보상체계 마련)을 통한 안전운행 유도

라. 제도 운영을 위한 예상 문제점

- 안전운행 보상지급을 위한 지속적인 예산확보(적정수준의 보상 필요)
- 기존 운영 중인 관제시스템과 보상체계 연동(초기 투자 비용 발생)

2. 기존 ADAS 제품과 성능 비교 관련 자료 보고서 제시가 필요하며 국토부 ADAS 인증 계획 제시 필요

ADAS 제품 기능 비교

성능 및 기능	MDAS-9	Mobileye630	AXON3.2	UXDTG
Camera	HD	HD	HD	FHD
LDW(차선이탈)	✓	✓	✓	✓
FCW(전방충돌)	✓	✓	✓	✓
PCW(보행자충돌)	✓		✓	
SLR(제한속도표시)	✓	✓		
DVR(비디오 저장)	✓	✓	✓	✓
FVSA(전방차량출발)				
SDA(안전거리)				✓
다차선 인식				✓
GPS	✓		✓	✓

ADAS 제품 하드웨어 비교

Specification	MDAS-9	Mobileye630	AXON3.2	UXDTG
CPU	Cortex A7 Quad	EyeQ4	Cortex-A53 Octa core	Quad-core ARM A57
GPU	-	-	-	128-core Maxwell
Front Camera	HD	HD	HD	FHD
Rear Camera	HD	-	-	-
DVR spec	MicroSD 64GB	MicroSD 128GB	MicroSD 64GB	MicroSD 128GB
Input/Output	Micro 5pin USB GPS, VIDEO	-	-	USB HDMI Eth
Temperature	-20°C ~ 70°C	-	-	0°C~60°C

국토교통부 전방충돌경고장치 시험방법 및 평가방법

1. 용어정의

- 가. “전방충돌경고장치”란 주행 중인 차량의 전방에서 동일한 방향으로 이동 중이거나 정차중인 자동차를 감지하고, 앞차와의 충돌을 회피하거나 완화하기 위해 운전자에게 경보를 주기 위한 장치를 말한다.
- 나. “대상자동차”란 전방충돌경고장치가 장착되어 그 성능을 평가하기 위한 자동차를 말한다.
- 다. “목표자동차”란 전방충돌경고장치가 장착된 자동차에 선행하여 감지대상이 되는 자동차를 말하며 세단형의 중형 승용자동차 또는 더미자동차(모형)로 한다.

2. 일반사항

- 가. 전방충돌경고장치를 위한 경보를 위해서는 시각, 청각, 촉각, 또는 그 조합에 의한 수단을 사용할 수 있으며, 청각이나 촉각(진동, 제동 등)에 의한 수단을 최소한 1개 이상 포함하여야 한다.
- 나. 시험차에 적응순항제어장치(ACC, Adaptive Cruise Control)가 장착된 경우, 전방충돌경고장치관련 시험을 수행하는 동안 적응순항제어장치(ACC)에 의한 자동적인 제어가 발생하지 않도록 한다.
- 다. 전방충돌경고장치는 운전자가 충돌경고장치의 작동 스위치를 끈 경우(또는 70km/h 미만의 저속 주행시) 작동하지 않을 수 있다.
- 라. 충돌경고장치 시험을 위한 목표자동차에는 감지능력을 최대화하기 위해 번호판을 반드시 부착한다.
- 마. 전방충돌경고장치가 다른 모드에서 작동하여 이 기준에서 요구하는 성능을 만족하지 못할 경우에는, 이상적인 작동 상태가 아님을 운전자에게 알려주어야 한다.
- 바. 만약 운전자가 전방충돌경고장치의 성능을 조절할 수 있는 경우, 적어도 하나의 조정값에서 시험기준을 만족해야 한다.
- 사. 시험하는 동안 자동차의 변속기어는 다음과 같이 한다.
 - 1) 목표자동차가 정지상태인 경우의 시험(시험자가 탑승할 필요는 없음)은 변속기어를 중립위치로 하고 주차제동장치를 작동시키며, 시험하는 동안 제동등이 점등되지 않도록 한다.
 - 2) 목표자동차가 감속중인 경우 및 목표자동차가 저속주행중인 경우의 시험
 - 가) 자동변속기 : 중립
 - 나) 수동변속기 : 시험속도에 적합한 최고 기어단수(클러치 연결상태)

3. 시험조건

가. 시험노면 및 차선규격

- 1) 시험을 위한 노면은 직선의 건조하고 편평한 포장노면(아스팔트 또는 콘크리트)으로 한다.
- 2) 시험을 위해서는 2개의 흰색 차선표시가 있거나, 1개의 차선만 사용될 경우 시험하는 동안 대상자동차 및 목표자동차의 중심선이 차선의 내측으로부터 1.8m 거리에 위치하여야 한다.
- 3) 2개의 차선이 사용되는 경우 차선의 내측 사이 간격은 최소 3.66m 이상이어야 하며 시험차는 그 중앙에 위치한다.

나. 대기조건

- 1) 시험 중 대기온도는 0 ~ 38℃ 이내, 풍속은 10m/s 이내로 한다.
- 2) 시험을 위한 장소는 시계거리 5,000m 이상의 분명한 시야가 확보된 곳으로 하며, 시험차선 주변에 표지판이나 다리 등 장애물이 없어야 한다.
- 3) 태양이 수평면에 대해 15도 이하로 기운 상태에서 태양을 향하는 방향으로 시험을 실시하지 않도록 한다.

다. 시험장비·센서 및 교정

- 1) 시험을 위한 데이터취득장비의 무게는 최대 26kg 이하로 하며 가급적 2열(뒷쪽 승객석)에 장착한다.
- 2) GPS 위치·속도 센서의 무게는 최대 12kg 이하로 한다.
- 3) 데이터 취득을 위한 샘플링율은 100Hz 로 한다.
- 4) 신호처리를 위한 필터는 왜곡현상(aliasing)을 피하기 위해 2차 이상의 저주파여과 버터워스(low-pass Butterworth) 필터를 사용하며 차단주파수는 15Hz로 한다.
- 5) 데이터 취득은 각 시험모드별 시험시작 3초 이전부터 시작한다.
- 6) 속도센서는 접촉 또는 비접촉식 센서가 사용될 수 있으며 GPS 센서가 사용되는 경우 재생율(refresh rate)이 100Hz 이상이어야 한다.
- 7) 시험을 위해 사용되는 주요센서의 추천규격은 다음과 같다.

형 식	출 력	측정범위	정밀도	정확도
속도센서	속 도	0.1 ~ 100km/h	0.2km/h 이상	±0.25% FS
요우잉센서	요우잉율	±100deg/s	0.01deg/s	±0.25% FS
위치센서	종방향위치	0~1.8m	5cm	<10cm
	횡방향위치	0~1.8m	5cm	<10cm
경고감지센서	경고발생여부	0~10V 또는 CAN	-	반응속도 10ms 이내

※ 속도, 요우잉, 위치 등 측정을 위해 DGPS(Differentially corrected GPS)가 사용될 수 있음

- 8) 속도센서의 교정이나 교정값 확인을 위해 300m 이상의 거리를 직선주행한다.
- 9) 계측장비 설치가 완료된 시험차를 100km/h의 속도로 균일하고 건조한 시험로를 따라 직진주행하는 동안 모든 채널에서 5 ~ 15초 동안 데이터를 저장하여 이상유무를 확인한다.
- 10) 운전자에 대한 경고는 계측장비를 통해 계측되어야 하며, 계측 수단은 이산신호형태로 직접 기록하거나 CAN, 혹은 경고발생 시점을 정확히 기록하기 위한 다른 어떤 수단도 가능하다.

4. 시험방법

가. 목표자동차가 정지상태인 경우

1) 시험기준

- 가) 대상자동차가 정지상태인 목표자동차에 접근할 때 적어도 TTC(Time To Collision) 2.1초 이전에 충돌경고가 발생되어야 한다.
- 나) 시험은 모두 7회 실시하여 적어도 5회가 충돌경고기준을 만족하여야 하며, 2회 연속 기준을 만족하지 못하는 경우는 부적합한 것으로 본다. 만약, 처음 5회가 기준을 만족하는 경우, 다음 2회가 연속해서 부적합한지를 확인하기 위해 시험을 계속할 필요는 없다.

2) 시험방법

- 가) 목표자동차를 대상자동차와 같은 방향을 향하도록 주행차로의 중심선에 정차시키고 제동등은 점등되지 않도록 한다.
- 나) 대상자동차는 초기속도 72km/h로 목표자동차를 향하여 주행한다. 이때 대상자동차의 변속기어는 "주행(Drive)" 상태로 하고, 수동변속기인 경우 요구 속도를 낼 수 있는 최고 단수로 하여 시험을 종료할 때까지 연결상태에 있도록 한다.
- 다) 대상자동차가 목표자동차로부터 150m 간격으로 접근하였을 때 시험을 시작하여, 다음 중 하나의 조건이 충족되었을 때 종료한다.
 - (1) 규정된 전방충돌경고장치 경고가 발생하거나,
 - (2) 목표자동차까지의 TTC가 1.9초 밑으로 떨어졌을 때
- 라) 위 다)의 시험 종료조건이 충족된 후 대상자동차 운전자는 목표자동차와 충돌을 방지하기 위해 조향장치나 제동장치를 작동시킨다.
- 마) 위 시험을 수행하는 동안 다음 조건들이 충족되어야 한다.
 - (1) 대상자동차의 속도는 규정된 전방충돌경고장치 경고가 발생하거나 목표자동차까지의 TTC가 1.9초 밑으로 떨어지기 전 3초 동안 초기속도에서 $\pm 1.6\text{km/h}$ 를 초과 이탈하지 않도록 한다.
 - (2) 대상자동차 운전자는 규정된 전방충돌경고장치 경고가 발생하거나 목표자동차까지의 TTC가 1.9초 밑으로 떨어지기 전에 브레이크페달에 어떠한 힘도 가해서는 안된다.
 - (3) 시험차로에서 대상자동차의 중심선과 목표자동차의 중심선 사이의 횡방향 간격은 0.6m 이내여야 한다.
 - (4) 시험하는 동안 대상자동차의 요우잉율(yaw rate)은 $\pm 1\text{deg/sec}$ 를 초과할 수 없다.

나. 목표자동차가 감속중인 경우

1) 시험기준

- 가) 대상자동차가 감속중인 목표자동차에 접근할 때 적어도 TTC 2.4초 이전에 충돌경고가 발생되어야 한다.

나) 시험은 모두 7회 실시하여 적어도 5회가 충돌경고기준을 만족하여야 하며, 2회 연속 기준을 만족하지 못하는 경우는 부적합한 것으로 본다. 만약, 처음 5회가 기준을 만족하는 경우, 다음 2회가 연속해서 부적합한지를 확인하기 위해 시험을 계속할 필요는 없다.

2) 시험방법

가) 대상자동차와 목표자동차는 진행방향으로 차간 중심거리를 30m로 유지하며 72km/h의 초기속도로 주행한다. 이때 대상자동차와 목표자동차의 변속기어는 "주행(Drive)" 상태로 하고, 수동변속기인 경우 시험 종료시까지 요구 속도를 낼 수 있는 최고 단수로 유지한다.

나) 목표자동차가 감속(제동장치를 급격히 작동시킨 후 1.5초 이내에 0.3g의 일정 감속도 유지)을 시작하기 약 7초 전에 시험을 시작하여, 다음 중 하나의 조건이 충족되었을 때 종료한다.

- (1) 규정된 전방충돌경고장치 경고가 발생하거나,
- (2) 목표자동차까지의 TTC가 2.2초 밑으로 떨어졌을 때

다) 위 나)의 시험 종료조건이 충족된 후 대상자동차 운전자는 목표자동차와 충돌을 방지하기 위해 조향장치나 제동장치를 작동시킨다.

라) 위 시험을 수행하는 동안 다음 조건들이 충족되어야 한다.

- (1) 목표자동차의 속도는 목표자동차의 감속이 시작되기 전 3초 동안 초기속도에서 $\pm 1.6\text{km/h}$ 를 초과 이탈하지 않도록 한다.
- (2) 대상자동차의 속도는 규정된 전방충돌경고장치 경고가 발생하거나 목표자동차까지의 TTC가 2.2초 밑으로 떨어지기 전 3초 동안 초기속도에서 $\pm 1.6\text{km/h}$ 를 초과 이탈하지 않도록 한다.
- (3) 시험차로에서 대상자동차의 중심선과 목표자동차의 중심선 사이의 횡방향 간격은 0.6m 이내여야 한다.
- (4) 시험하는 동안 대상자동차 및 목표자동차의 요우잉율(yaw rate)은 $\pm 1\text{deg/sec}$ 를 초과할 수 없다.
- (5) 목표자동차의 감속도는 제동시작 후 1.5초 이내에 0.3g로 유지하여야 한다. 목표자동차의 감속도 허용가능오차는 전방충돌경고장치 경고가 최초 발생한 시점에 측정하였을 때 $\pm 0.03\text{g}$ 이다. 0.3g를 초과하는 초기 오버슈트는 허용되나 0.05초 이상의 시간 동안 0.375g를 초과하여서는 안되며, 최초의 감속도 최대치 도달 후 0.5초를 지난 시점부터 전방충돌경고장치 경고가 발생하였을 때까지 0.33g를 초과하여서는 안된다.
- (6) 진행방향으로 대상자동차로부터 목표자동차까지 차간 중심간격의 허용오차는 목표자동차 제동시작 전 3초 시점과 목표자동차 제동시작 시점에서 측정하였을 때 $\pm 2.5\text{m}$ 이내이어야 한다.
- (7) 대상자동차 운전자는 규정된 전방충돌경고장치 경고가 발생하거나 목표자동차까지의 TTC가 2.2초 밑으로 떨어지기 전에 브레이크페달에 어떠한 힘도 가해서는 안된다.

다. 목표자동차가 저속주행중인 경우

1) 시험기준

가) 대상자동차가 감속중인 목표자동차에 접근할 때 적어도 TTC 2.0초 이전에 충돌경고가 발생되어야 한다.

나) 시험은 모두 7회 실시하여 적어도 5회가 충돌경고기준을 만족하여야 하며, 2회 연속 기준을 만족하지 못하는 경우는 부적합한 것으로 본다. 만약, 처음 5회가 기준을 만족하는 경우, 다음 2회가 연속해서 부적합한지를 확인하기 위해 시험을 계속할 필요는 없다.

2) 시험방법

- 가) 대상자동차는 초기속도 72km/h로 150m 이상 충분한 간격을 두고 목표자동차를 향해 주행하고, 목표자동차는 32.2km/h의 일정 속도로 동일한 방향으로 주행한다. 이때 대상자동차와 목표자동차의 변속기어는 "주행(Drive)" 상태로 하고, 수동변속기인 경우 시험 종료시까지 요구 속도를 낼 수 있는 최고 단수로 유지한다.
- 나) 진행방향으로 대상자동차로부터 목표자동차까지의 차간 중심거리가 100m가 되었을 때 시험을 시작하여, 다음 중 하나의 조건이 충족되었을 때 종료한다.
- (1) 규정된 전방충돌경고장치 경고가 발생하거나,
 - (2) 목표자동차까지의 TTC가 1.8초 밑으로 떨어졌을 때
- 다) 위 나)의 시험 종료조건이 충족된 후 대상자동차 운전자는 목표자동차와 충돌을 방지하기 위해 조향장치나 제동장치를 작동시킨다.
- 라) 위 시험을 수행하는 동안 다음 조건들이 충족되어야 한다.
- (1) 대상자동차의 속도는 규정된 전방충돌경고장치 경고가 발생하거나 목표자동차까지의 TTC가 1.8초 밑으로 떨어지기 전 3초 동안 초기속도에서 $\pm 1.6\text{km/h}$ 를 초과 이탈하지 않도록 한다.
 - (2) 목표자동차의 속도는 시험하는 동안 $32.2 \pm 1.6\text{km/h}$ 를 초과 이탈하지 않도록 한다.
 - (3) 시험차로에서 대상자동차의 중심선과 목표자동차의 중심선 사이의 횡방향 간격은 0.6m 이내여야 한다.
 - (4) 시험하는 동안 대상자동차 및 목표자동차의 요우잉율(yaw rate)은 $\pm 1\text{deg/sec}$ 를 초과할 수 없다.
 - (5) 대상자동차 운전자는 규정된 전방충돌경고장치 경고가 발생하거나 목표자동차까지의 TTC가 1.8초 밑으로 떨어지기 전에 브레이크페달에 어떠한 힘도 가해서는 안된다.

국토교통부 차로이탈경고장치 시험방법 및 평가방법

1. 용어정의

- 가. "차로이탈경고장치(LDWS, Lane Departure Warning System)"는 자동차가 주행하는 차로를 운전자의 의도와는 무관하게 벗어나는 것을 운전자에게 경고하는 장치를 말한다.
- 나. "차로"는 자동차가 도로의 정해진 부분을 한 줄로 통행할 수 있도록 차선에 의해 구분되는 차도의 부분을 말한다.
- 다. "차선"은 자동차가 주행하는 동안 운전자에 의하여 직접적으로 식별될 수 (예를 들어 눈으로 덮이지 않는 등) 있도록 차로의 경계선에 계획적으로 설치한 (반사식)도로 경계 표시를 말한다.
- 라. "이탈 속도"는 경고발생지점에서의 차선에 대해 직각방향으로 대상자동차가 접근하는 속도를 말한다.
- 마. "공유구역"이라 함은 2개 이상의 식별표시, 표시장치, 식별부호 또는 그 밖의 메시지를 표시하지만 동시에 표시하지 아니하는 구역을 말한다.
- 바. "추천냉간팽창공기압"이란 해당 자동차의 속도 및 하중 등 사용조건에 따라 제작자가 각 위치의 타이어에 대해 권장하는 타이어공기압(타이어공기압표찰 또는 취급설명서에 표시한다)을 말한다.
- 사. "경적차상태"란 공차상태에서 운전자 외의 시험장비를 포함하는 상태를 말한다. 다만, 앞좌석에 시험 결과를 기록할 수 있는 측정자 1명을 추가하여도 된다.
- 아. "대상자동차"란 차로이탈경고장치가 장착되어 그 성능을 평가하기 위한 자동차를 말한다.

2. 시험조건

가. 시험차로

- 1) 시험은 폭 $3.5\text{m} \pm 0.01\text{m}$ 의 차로에서 실시한다.
- 2) 시험을 위해 설치되는 시험차선의 종류 및 규격은 그림1과 같으며, 차선은 “도로교통법 시행규칙 제 8조제2항 및 교통노면표시 설치·관리매뉴얼(경찰청)”에 적합한 재질로 제작되어야 하고, 우수한 상태이어야 한다.

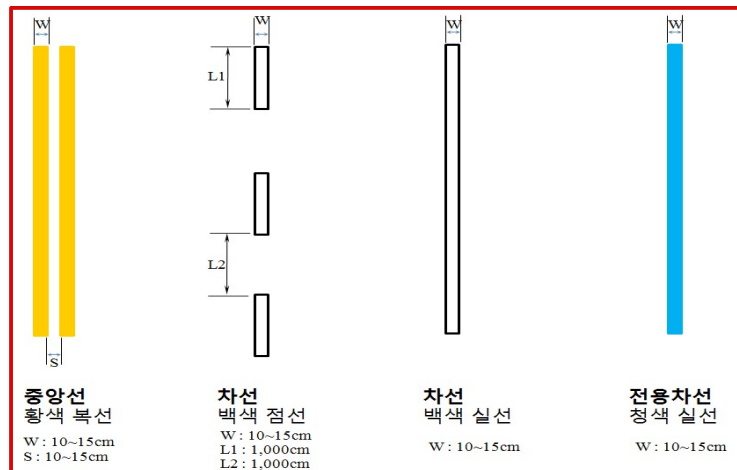


그림 1 시험차선의 종류 및 규격

- 3) 위 2)에 따라 설치된 시험차선 표시는 시인성이 양호한 상태이어야 한다.
- 4) 시험은 평탄하고 건조한 아스팔트 또는 콘크리트 노면에서 실시한다.

나. 대기조건

- 1) 시험중 대기 온도는 $0 \sim 45^{\circ}\text{C}$ 이내, 풍속은 최대 10m/s 이내로 한다.
- 2) 비, 눈, 우박, 안개, 연무, 화산재 등의 악조건의 기상상태에서는 시험을 실시하지 않으며, 낮시간 및 양호한 기상 조건에서 시험을 실시한다.
- 3) 태양이 수평면에 대해 15° 이하로 기운 상태에서 태양을 향하는 방향으로 시험을 실시하지 않도록 한다.

다. 자동차 조건

- 1) 시험자동차는 전차축 또는 후차축 설계허용하중을 초과하지 않고 제작자가 정한 차축간의 하중배분에 따른 경적차상태 또는 적차상태로 실시한다. 다만, 시험 중에는 하중조건을 변경할 수 없다.
- 2) 시험 중 시험 자동차의 주행을 위해 사용하는 변속기 단수는 자동변속기인 경우 “D”단을 사용하고, 수동 변속기인 경우 시험에 요구되는 속도를 유지할 수 있는 최고 단수를 사용한다.
- 3) 시험 자동차의 타이어 공기압은 추천냉간팽창공기압으로 한다.
- 4) 사용자가 경고한계를 설정할 수 있는 차로이탈장치는 차로이탈경고시험 시 경고이탈거리를 최대로 설정하여 실시하고, 시험중에는 변경하지 않아야 한다.

3. 시험방법

- 가. 시험자동차의 자세가 안정되도록 순조롭게 시험차로의 가상의 중심선을 따라 시험속도를 시속 65킬로미터로 유지시켜 주행시킨다, 이탈경고가 발생될 때까지 시험속도의 허용오차는 $\pm 3\text{km/h}$ 로 한다.
- 나. 시험자동차의 시험속도를 유지하며 임의의 조향핸들각을 입력하여 점진적으로 0.1m/s 내지 0.8m/s사이의 이탈속도로 좌측 또는 우측으로 주행 중인 차로를 벗어나게 한다.
- 다. 이탈하는 방향의 차선에 가장 가까운 시험자동차의 전륜타이어 외측이 차선의 외측모서리로부터 0.3m 이상 까지를 가로지르게 하고, 이때 아래 4호의 각목에 규정된 이탈경고를 제공하는지 여부 및 경고 시점에서의 차선의 외측 모서리와 타이어 외측과의 거리를 확인한다. 이때, 차선의 외측 모서리와 전륜 휠의 타이어 외측과의 거리 측정 위치는 아래의 그림 2와 같이 타이어 접지면 외측을 기준으로 측정한다.

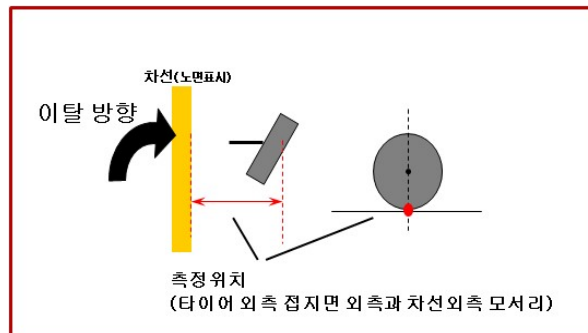


그림 2 차로이탈경고장치의 경고 시점 위치 측정

- 라. 앞서 시행한 이탈속도와 다른 이탈속도로 가.목 내지 다.목을 반복한다.
- 마. 앞서 시행한 방향과 반대인 방향으로 가.목 내지 라.목의 시험을 반복한다.

4. 평가방법

차로이탈경고장치의 성능은 아래의 기준을 만족하는 경우 고시된 점수를 부여한다.

가. 성능기준

- 1) 차로이탈경고장치는 운전자가 수동으로 차로이탈경고장치의 기능을 해제시키지 않는 한 최소한 시속 60킬로미터 이상에서 작동되어야 한다.
- 2) 차로이탈경고는 아래의 가) 또는 나)를 사용하여 경고하여야 한다.
 - 가) 시각, 청각 또는 촉각경고 중에서 2개 이상의 수단
 - 나) 자동차가 차로를 이탈하는 방향을 운전자가 알 수 있도록 하는 촉각 또는 청각 경고 중 어느 하나의 수단(이 경우 제공되는 경고는 실제 이탈하는 방향과 일치할 것)
- 3) 그림 2와 같이 측정하였을 때, 차선에 가장 가까운 시험자동차의 전륜타이어 접지면 외측이 이탈되는 방향의 차선 바깥쪽 모서리에서 0.3m 떨어진 가상의 선이내에서 위 2)에 규정된 차로이탈경고를 제공하여야 한다.

나. 성공률기준

차로이탈경고장치는 위의 2. 시험조건 및 3. 시험방법을 만족하는 시험 총 횟수의 90%이상 4호가목2)를 만족하여야 한다.

국토부 ADAS 인증 계획

국토교통부 전방충돌경고 및 차로이탈 인증을 받기 전 개발된 기능들을 아래와 같은 테스트 프로세스를 수행하여 기술 고도화 및 기능 보완

국토교통부 전방충돌경고 및 차로이탈경고 성능기준을 테스트하기 위해 여러기관에 (SOC실증연구센터-연천-KICT, 지능형자동차부품진흥원-KIAP, 자동차융합기술원-JIAT) 문의 후 사전 테스트 진행



한국자동차연구원-KATECH에서 FCWS, LDWS 시험검사의뢰

3. 본 기술의 한계점, 기존 기술과의 차별성 및 우수성 등 제시 필요(예: 영상기반 전방 충돌경보 기능의 기존 기술 대비 특징점 등)

3.1. 기술의 한계점

- 전체 구성 기술의 핵심은 차량운행기록장치, OBD, ADSA, GPS, 블록체인 보상시스템으로 구성
- 현재 기관에서 제공되는 GPS 정보를 기반으로 보상구간을 특정하기가 불가능, DGPS 도입 필요
- 블록체인 보상시스템 개발, 운영, 관리를 위해서는 대규모 투자가 필요
- 블록체인 데이터처리를 위한 실시간성 확보를 위한 인프라의 투자 필요(5G)
- 본 연구를 통해 확보한 ADAS 기술을 기존 검증된 ADAS 상용품에 추가 적용하여 상용화 필요

3.2. 기존 기술과의 차별성 및 우수성

- 기존 ADAS 상용품에 없는 중앙선 침범, 규정차선인식, 불법유턴 알고리즘 개발 구현
- 차량운행기록장치(DTG)+ADAS+블록체인 보상시스템이 통합된 시스템 전세계 최초 개발
- 차량운행기록장치(DTG)의 데이터 위변조를 방지하기 위해 개발된 라이트 블록체인 기술은 통신이 두절된 상태에서도 위변조가 불가능하도록 개발됨(SCI 논문 게재 / electronics. 2021. 03)
- 블록체인 데이터 처리 속도를 향상하기 위해 ‘블록체인 데이터처리 가속 알고리즘 개발’ 하여 블록 읽기 성능은 연구초기 12.51TPS에서 1,040TPS로 **83배 이상 향상**, 블록 삭제의 경우 연구초기 9.21TPS에서 1,863.97TPS로 **200배 이상의 성능향상**.(인터넷정보학회논문 게재. 2022. 03)

4. 사업화 전략 및 활용방안에 대한 구체적 명시 필요

4.1. 사업화 전략 및 활용방안

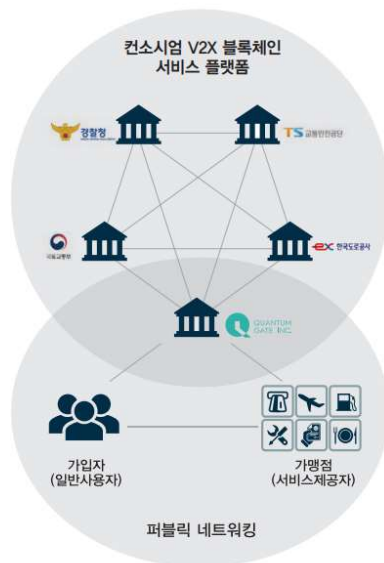
가. 제품 개발

- 기존 ADAS 상용품 업체와 NDA, MOU를 체결하여 당사 알고리즘 상용화 제품에 추가 적용 개발
- 상기 제품 국토부 ADAS 인증 추진
- 교통안전공단(DTG) 및 국토부 표준 규격(ADAS) 등을 적용하여 제품 상용화 개발 및 인증획득

나. 공공부문

- 전국화물운송조합, 버스운송조합과 컨소시엄을 구성 국토부 시범사업 추진
- 교통사고 감소, 보험료 지급 감소, 교통법규 위반 횟수 감소 등의 효과검증 확보
- 교통안전 공단 차량운행 기록 데이터의 위변조 방지 데이터 제공
- 차량운행기록장치의 새로운 표준으로 확산

블록체인 기반의 V2X 플랫폼 서비스 체계 (구상 예시)



[컨템게이트]

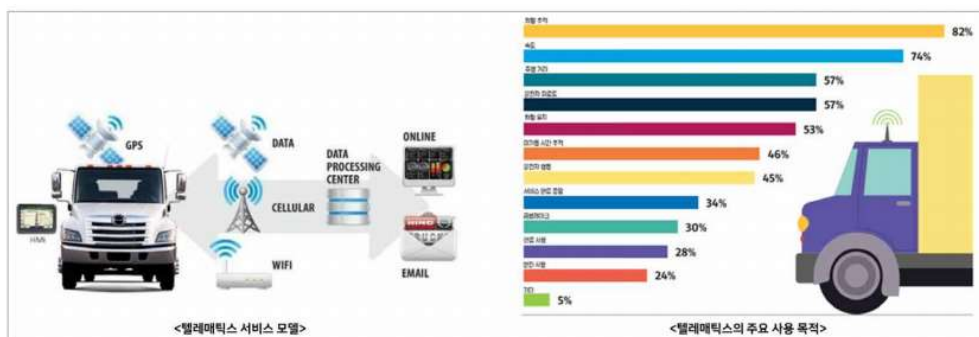
V2X 블록체인 네트워크를
교통관련 정부 기관과 공유하고
시스템 설계, 유지, 관리

컨소시엄 블록체인

- 반 중앙형 블록체인
- 미리 선정된 소수(N개)의 주체들만 참여 가능
- 주체들간 합의된 규칙을 통해 공증 참여
- 네트워크 확장이 용이하고 거래속도 빠름
- 활용사례 : R3 CEV, HSBC, Citi, Barclays 등

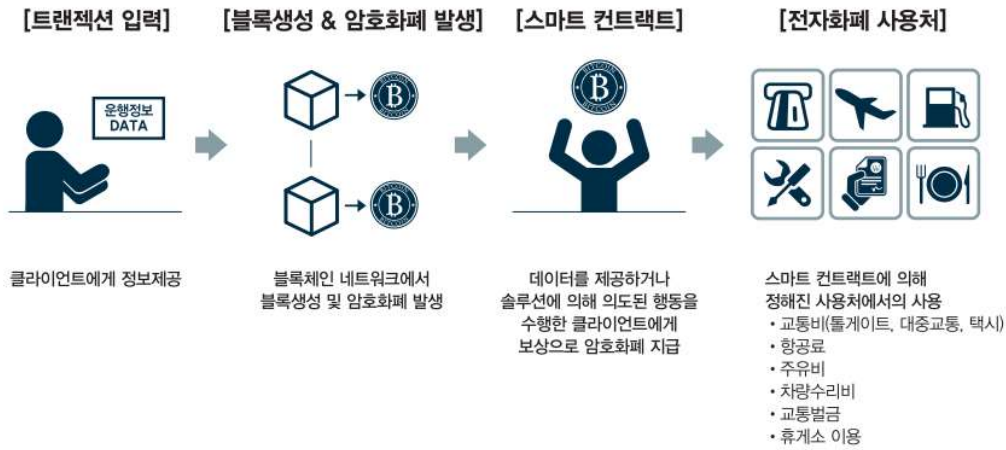
다. 민간부문 확산

- 전국화물운송조합, 버스운송조합 차량운행기록장치 제품 설치 권장
- 대형 운송법인 및 유통회사(CJ 대한통운, SPC그룹, 이마트 등) 납품
- 특수화물 차량(시멘트, 탱크로리, 컨테이너, 정유차량, 의약품, 농수산물, 화학약품 등)
- 택시, 공유 카 플랫폼 서비스 업체 등 텔레매틱스 분야 진출 확대



라. 기타 수익 모델

- 단말기 판매 : 디지털운행기록장치+ADAS(전방추돌, 차선이탈)+안전운행 보상시스템
- 각 사업자별 특화된 차량관제 시스템 제공으로 시스템 판매, 교육, 유지보수 수익
- 블록체인 안전운행보상 스마트컨트랙트 시스템운영 관리 수익
- 전자바우처 생태계 조성을 통한 전자바우처 거래 수수료



5. 블록체인 보상시스템 보상/패널티에 기준설정 방안이나 보상시스템 운영 관계기관과의 협의계획 제시 필요

5.1. 블록체인 보상시스템 보상/패널티에 기준설정 방안

- ❶ 현재 위험운전행동은 운행기록 분석 시스템에서 제공하고 있으며, 11가지 위험운전행동으로 정의

	내용	범위	변수형	크기
0	년월일	YYMMDD	unsigned int	4byte
1	GPS시간	hhmmss.cccc	unsigned float	4byte
2	위험운전행동	과속 장기과속 급가속 급출발 급감속 급정지 급좌회전 급우회전 급U턴 급앞지르기 급진로변경	unsigned char	1byte

- ❷ 본 연구를 통해 확보된 운행데이터 및 영상기반 데이터로 하기와 같은 보상 및 패널티 기준 설정

	내용	범위	변수형	크기
0	년월일	YYMMDD	unsigned int	4byte
1	GPS시간	hhmmss.cccc	unsigned float	4byte
2	법규위반 행동	핸들과다조작 신호위반 불법유턴 중앙선침범 지정차로위반 차로 급 변경 안전거리 미확보	unsigned char	1byte

- ❸ 하기의 내용에 근거하여 보상 및 패널티 설정 기준으로 함.

- 위험운전행동(DTG 정보)
과속, 장기과속, 급가속, 급출발, 급감속, 급정지, 급좌회전, 급우회전, 급U턴, 급앞지르기, 급진로변경
- 법규위반행동(ADAS 정보)
신호위반, 불법유턴, 중앙선침범, 지정차로위반, 차로 급 변경, 안전거리 미확보

5.2. 보상시스템 운영 관계기관과의 협의 계획

- 관계기관 협의 대상, 교통안전공단(차량운행기록장치 데이터 관리기관)
- 본 연구과제의 행정절차가 최종 마무리되면 교통안전공단과 협의 예정(22. 5월 중)

< 별첨 >

별첨 자료

1. 참고문헌

2. 총 연구개발기간 성과점검기준표 및 증빙자료

3. 세부 정량적 연구개발성과 증빙자료(총 연구개발기간 성과점검기준표 실적 증빙자료에 기 포함되어 있는 증빙자료 제외)

[SCI 논문 References]

1. OECD Data, Road Accidents. Available online: <https://data.oecd.org/transport/road-accidents.htm> (accessed on 9 February 2021).
2. Joint Research Center Digital Tachograph. Available online: <https://dtc.jrc.ec.europa.eu/> (accessed on 3 December 2020).
3. ELD Mandatory for Land Trucks in US, Increase of Distribution Costs. Available online: http://www.centralx.co.kr/bbs/board.php?bo_table=notice&wr_id=21&sfl=&stx=&sst=wr_hit&sod=asc&sop=and&page=1 (accessed on 1 December 2020).
4. Management Guidelines for Automobile Operation Records and Devices. Korea Ministry of Land, Infrastructure and Transport Announcement No. 2020-273. Available online: <http://www.molit.go.kr/USR/policyTarget/dtl.jsp?idx=311> (accessed on 1 December 2020).
5. Traffic News. Voices Raised to Reexamine DTG Operation Record Transmission System. Available online: http://www.bstci.or.kr/xs/index.php?mid=safy_info_board&document_srl=62790 (accessed on 27 July 2020).
6. Baek, S.; Sohn, M. Design and implementation of stable data storage for digital tachographs. *Korea Inf. Process. Soc. J.* 2012, 1, 71–78. [CrossRef]
7. Park, G. Design and Implementation of TPM-Based Tamper-Resistant Digital Tachographs. Available online: <http://syscore.sejong.ac.kr/~{woongbak/publications/DJ12.pdf> (accessed on 31 October 2020).
8. Kim, G. Korea Patent No.1020954340000. Korean Intellectual Property Office. Available online: <https://doi.org/10.8080/1020180109087> (accessed on 22 July 2020).
9. ATEC T & Co., Ltd. Intelligent Driver Assistance System Equipped with Accident Video Transmission Function. Advanced Driver Assistance Aystem with Accident Video Transmission. Korea Patent No. 1020180138027, 12 November 2018. [CrossRef]
10. Nakamoto, S. Bitcoin P2P e-Cash Paper. Available online: <https://archive.is/DUcfX> (accessed on 31 October 2020).
11. Lightcoin Foundation. 2020. Available online: <https://litecoin-foundation.org/about-us/> (accessed on 1 December 2020).
12. Ethereum Foundation. Available online: <https://ethereum.org/en/foundation/> (accessed on 5 February 2020).
13. Ripple Company. Available online: <https://ripple.com/company/> (accessed on 5 December 2020).
14. BitcoinCash.Org. Available online: <https://www.bitcoincash.org/> (accessed on 5 December 2020).
15. EOS. Available online: <https://eos.io/> (accessed on 4 December 2020).
16. Poon, J.; Dryja, T. The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments. Available online: <https://lightning.network/lightning-network-paper.pdf> (accessed on 5 December 2020).
17. Poon, J.; Buterin, V. Plasma: Scalable Autonomous Smart Contracts. Available online: <https://plasma.io/plasma.pdf> (accessed on 5 December 2020).
18. Hong, J. Side Chain the Solution to Blockchain Scalability. Available online: <https://medium.com/@joycehong0524/blockchain-scalability-solution-side-chain-feat-plasma-f3a530cd482c> (accessed on 5 December 2020).
19. Dai, P.; Mahi, N.; Earls, J.; Norta, A. Smart-Contract Value-Transfer Protocols on a Distributed Mobile Application Platform. Qtum Whitepaper, Qtum Foundation. Available online: https://qtum.org/user/pages/01.home/Qtum%20New%20Whitepaper_en.pdf (accessed on 5 December 2020).
20. Nem Group. Available online: <https://nem.io/about> (accessed on 5 December 2020).
21. Hdac Technology AG. Available online: <https://www.hdactech.com/en/company/about.do> (accessed on 5 February 2021).
22. Liu, Y.; Wang, K.; Lin, Y.; Xu, W. LightChain: A Lightweight Blockchain System for Industrial Internet of Things. *IEEE Trans. Ind. Inform.* 2019, 15, 3571–3581. [CrossRef]
23. Bai, L.; Hu, M.; Liu, M.; Wang, J. BPllot: A Light-Weighted Blockchain-Based Platform for Industrial IoT. *IEEE Access* 2009, 7, 58381–58393. [CrossRef]
24. Gruber, D.; Li, W.; Karame, G. *Unifying Lightweight Blockchain Client Implementations*; Workshop on Decentralized IoT Security and Standards (DISS): San Diego, CA, USA, 2018.
25. Lin, T.; Yang, X.; Wang, T.; Peng, T.; Xu, F.; Lao, S.; Ma, S.; Wang, H.; Hao, W. Implementation of High-Performance Blockchain Network Based on Cross-Chain Technology for IoT Applications. *Sensor* 2020, 11, 3268. [CrossRef] [PubMed]
26. Yue, L.; Junqin, H.; Shengzhi, Q.; Ruijin, W. Big data model of security sharing based on blockchain. In Proceedings of the 2017 3rd International Conference on Big Data Computing and Communications (BIGCOM), Chengdu, China, 10–11 August 2017; pp. 117–121.
27. Xiong, Z.; Zhang, Y.; Niyato, D.; Wang, P.; Han, Z. When mobile blockchain meets edge computing. *IEEE Commun. Mag.* 2018, 56, 33–39. [CrossRef]
28. Yuan, Y.; Wang, F. Towards blockchain-based intelligent transportation systems. In Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, 1–4 November 2016; pp. 2663–2668.
29. Kang, J.; Yu, R.; Huang, X.; Wu, M.; Maharjan, S.; Xie, S.; Zhang, Y. Blockchain for Secure and Efficient Data Sharing in Vehicular Edge Computing and Networks. *IEEE Internet Things J.* 2019, 6, 4660–4670. [CrossRef]
30. Didouh, A.; Lopez, A.B.; Hillali, Y.E.; Rivenq, A.; Faruque, M.A.A. Eve, You Shall Not Get Access! A Cyber-Physical Blockchain Architecture for Electronic Toll Collection Security. In Proceedings of the 2020 IEEE 23th International Conference on Intelligent Transportation Systems (ITSC), Rhodes, Greece, 20–23 September 2020; pp. 1–7. [CrossRef]

-
31. Lee, S.; Jung, H.; Han, B. Security Assured Vehicle Data Collection Platform by Blockchain: Service Provider's Perspective. In Proceedings of the 2019 21th International Conference on Advanced Communication Technology (ICACT), PyeongChang, Gangwon-Do, Korea, 17–20 February 2019; pp. 265–268. [CrossRef]
 32. Bera, B.; Saha, S.; Das, A.K.; Kumar, N.; Lorenz, P.; Alazab, M. Blockchain-Envisioned Secure Data Delivery and Collection Scheme for 5G-Based IoT-Enabled Internet of Drones Environment. *Proc. IEEE Trans. Veh. Technol.* **2020**, *69*, 9097–9111. [CrossRef]
 33. Saldamli, G.; Karunakaran, K.; Vijaykumar, V.K.; Pan, W.; Puttarevaiah, S.; Ertaul, L. Securing Car Data and Analytics using Blockchain. In Proceedings of the IEEE 2020 Seventh International Conference on Software Defined Systems (SDS), Paris, France, 20–23 April 2020; pp. 153–159. [CrossRef]
 34. Lei, A.; Cruickshank, H.; Cao, Y.; Asuquo, P.; Ogah, C.P.A.; Sun, Z. Blockchain-Based Dynamic Key Management for Heterogeneous Intelligent Transportation Systems. *IEEE Internet Things J.* **2017**, *4*, 1832–1843. [CrossRef]
 35. Oracle. Available online: https://docs.oracle.com/cd/E17236_01/epm.1112/cb_statistical/frameset.htm?ch07s02s10s01.html (accessed on 2 December 2020).

국내 인터넷정보학회논문지 논문 참고문헌

- [1] hyperledger fabric <https://hyperledger-fabric.readthedocs.io/en/latest/index.html>, 2021.12.28.
 - [2] 사업용 차량 디지털운행기록계 장착 의무화, 국토해양부 보도자료, 2010.6.21.
 - [3] Nakamoto, Satoshi "Bitcoin P2P e-cash paper", Archived from the original on 28 December 2012. [https://archive.is/DUcfX\(31 October 2008\)](https://archive.is/DUcfX(31%20October%202008)).
 - [4] <https://blog.litecoin.org/>
 - [5] <https://ethereum.org/en/>
 - [6] <https://ripple.com/>
 - [7] <https://www.bitcoincash.org/>
 - [8] <https://eos.io/>
 - [9] 박정숙, 박준영, 최선미. 블록체인 세대별 기술 동향, 전자통신동향분석, Vol 33, NO.6, pp139–153, 2018.
 - [10] Su-Hwan Bae, Sun-Ok Cho, Yong-Tae Shin, "A System Recovery Using Hyper-ledger Fabric Blockchain", Vol12, No.2, 2019.
 - [11] Shafkat Islam, Shahriar Badsha, Shamik Sengupta, "A Light-weight Blockchain Architecture for V2V", 2020 IEEE International Smart Cities Conference (ISC2), 2020. 9.
 - [12] Yongbae Kim, Juyong Back, Jongweon Kim, "A Tamper-Resistant Algorithm Using Blockchain for the Digital Tachograph", Electronics, 2021
 - [13] Caliper Benchmarks, "Hyperledger Blockchain Performance Reports", <https://hyperledger.github.io/caliper-benchmarks>, 2022
-

주 의

1. 이 보고서는 국토교통부에서 시행한 국토교통과학기술 연구개발사업 ‘화물차 운전자(차량)의 안전운행 행위에 대한 보상을 위한, 블록체인 기반 디지털운행기록단말 장치와 안전운행 보상 시스템 개발’ 연구개발과제 최종보고서이다.
2. 이 연구개발내용을 대외적으로 발표할 때에는 반드시 국토교통부(국토교통과학기술진흥원)에서 시행한 국토교통과학기술 연구개발사업의 결과임을 밝혀야 한다.
3. 국가과학기술 기밀 유지에 필요한 내용은 대외적으로 발표 또는 공개하여서는 안 된다.