**Software Requirement Specification  
for Simple ADAS Driving Vehicle**

**Author**

**박성원**

Date

**2023-03-24**

**Team Information**

**개인 프로젝트**

**Table of Contents**

[1 개요 4](#_Toc127572881)

[1.1 목적 4](#_Toc127572882)

[1.2 범위 4](#_Toc127572883)

[1.3 용어 정리 4](#_Toc127572884)

[1.4 참고 문헌 4](#_Toc127572885)

[1.5 Overview 4](#_Toc127572886)

[2 개발 대상 설명 4](#_Toc127572887)

[2.1 개발 대상 4](#_Toc127572888)

[2.2 기능 (Statements of purpose) 4](#_Toc127572889)

[2.3 동작 모드 (Operating Mode) 4](#_Toc127572890)

[2.4 사용자 특징 5](#_Toc127572891)

[2.5 제약 및 가정 사항 5](#_Toc127572892)

[2.6 HW 사양 5](#_Toc127572893)

[3 세부 요구사항 5](#_Toc127572894)

[3.1 외부 인터페이스 5](#_Toc127572895)

[3.1.1 사용자 인터페이스 5](#_Toc127572896)

[3.1.2 HW 인터페이스 5](#_Toc127572897)

[3.1.3 SW 인터페이스 5](#_Toc127572898)

[3.1.4 통신 인터페이스 5](#_Toc127572899)

[3.2 기능 요구사항 5](#_Toc127572900)

[3.3 성능 요구사항 6](#_Toc127572901)

[3.4 설계 제약사항 6](#_Toc127572902)

[3.5 소프트웨어 속성 6](#_Toc127572903)

[3.6 기타 요구사항 6](#_Toc127572904)

1. 개요
   1. 목적

본 문서는 자율주행 RC카 제작 프로젝트를 위해 ADAS 기능을 간단한 소프트웨어로 개발하고 해당 기능의 요구사항을 명세한다. 개발한 소프트웨어는 Arduino와 같은 MCU를 장착한 RC카에 내장하여 테스트를 통해 기능의 정상 작동 여부를 확인한다. 본 프로젝트는 ADAS의 작동 원리를 이해 및 구현하고 소프트웨어 개발 프로세스를 적용하여 모델과 임베디드의 테스트 결과를 비교하는 것에 목적을 둔다.

* 1. 범위

기능 요구사항, 인터페이스, 개발 대상 등 요구사항 문서의 범위를 쓰는 챕터.

* 1. 용어 정리

HW: Hardware, 하드웨어

SW: Software, 소프트웨어

MCU: Micro Controller Unit, 마이크로 컨트롤러 유닛

VCU: Vehicle Control Unit, RC카 제어 유닛

ADAS: Advanced Driver Assistance System, 첨단 운전자 보조 시스템

LFA: Lane Following Assist, 차로 유지 보조

ACC: Adaptive Cruise Control, 적응형 순항 제어

FCA: Forward Collision-Avoidance Assist, 전방 충돌 방지 보조

BCA: Blind-Spot Collision-Avoidance Assist, 후측방 충돌방지 보조

SEA: Safety Exit Assist, 안전 하차 보조

ISLA: Intelligent Speed Limit Assist ,지능형 속도 제한 보조

DAW: Driver Attention Warning, 운전자 주의 경고

BVM: Blind-Spot View Monitor, 후측방 모니터

SCC: Smart Cruise Control, 스마트 크루즈 컨트롤

NSCC: Navigation-Based Smart Cruise Control, 네비게이션 기반 스마트 크루즈 컨트롤

LKA: Lane Keeping Assist, 차로 이탈방지 보조

HDA: Highway Driving Assist, 고속도로 주행 보조

HDA2: Highway Driving Assist 2, 고속도로 주행 보조 2

SVM: Surround View Monitor, 서라운드 뷰 모니터

RCCA: Rear Cross-Traffic Collision-Avoidance Assist, 후방 교차 충돌방지 보조

PCA: Parking Collision-Avoidance Assist, 후방 주차 충돌방지 보조

RSPA: Remote Smart Parking Assist, 원격 스마트 주차 보조

LAS: LFA의 Active State(동작 상태)

AAS: ACC의 Active State(동작 상태)

FAS: FCA의 Active State(동작 상태)

TTC: Time To Collision, 예상 충돌 시간

* 1. 참고 문헌

RC Car Kit:

<https://ko.aliexpress.com/item/1005001314610613.html?_randl_currency=KRW&_randl_shipto=KR&src=google&memo1=freelisting&aff_fcid=6f8a16bed5444ceb9e343621174889e5-1675603764141-02992-UneMJZVf&aff_fsk=UneMJZVf&aff_platform=aaf&sk=UneMJZVf&aff_trace_key=6f8a16bed5444ceb9e343621174889e5-1675603764141-02992-UneMJZVf&terminal_id=781d12290187498281b62950804ef4d0&afSmartRedirect=y&gatewayAdapt=Pc2Msite>

Arduino UNO R3 datasheet:

<https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf>

MPU-6050 datasheet:

<https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>

11kg MG996R Servo Motor:

<https://www.falconshop.co.kr/shop/goods/goods_view.php?goodsno=100005086>

GB37-520 Motor datasheet:

HC-SR04 Ultra Sensor datasheet:

<https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>

Genesis G80 User Manual for ADAS features:

<https://www.genesis.com/wsvc/kr/api/v2/download-center/pdfview>

1. 개발 대상 설명
   1. 개발 대상

본 프로젝트는 ADAS 차량 시스템의 구현을 목표로 하며, ADAS 기능의 일부를 모사한 RC카를 개발하고자 한다.

ADAS는 첨단 운전자 보조 시스템으로, 운전자가 안전하고 편리하게 주행할 수 있도록 도와주는 모든 기능을 뜻한다. ADAS의 기능으로는 FCA, LKA, BCA, SEA, ISLA, DAW, BVM, SCC, NSCC, LFA, HDA, HDA2, SVM, RCCA, PCA, RSPA 등이 있으나 본 프로젝트에서는 이 중 LFA, ACC, FCA만을 구현하는 것으로 기능의 수를 제한한다. 이 때, FCA는 제동만을 사용해 동작하는 것으로 기능을 제한한다.

해당 시스템은 판단부와 제어부의 2개의 서브 시스템으로 나눌 수 있다.

판단부는 PC, 블루투스 모듈, 카메라 모듈, 초음파 센서, IMU 센서, 아누이노 내 판단SW로 구성되어 있다. 판단부는 카메라 모듈, 초음파 센서, IMU 센서 데이터와 수신한 PC 입력을 바탕으로 LFA, ACC, FCA에 대한 계산 및 주행 판단을 수행하고 RC카의 목표 가속도와 목표 조향각을 제어부로 송신한다.

제어부는 PC, 블루투스 모듈, 아두이노 내 구동SW, IMU 센서로 구성되어 있다. 제어부는 판단부의 출력 값과 수신한 PC 입력을 바탕으로 전압을 계산하여 조향 서보 모터와 후륜 구동 모터를 작동시킨다.

우선 LFA를 구현하기위해 카메라 모듈을 사용하며, ACC, FCA를 구현하기 위해 카메라 모듈과 초음파 센서를 사용한다. LFA, ACC, FCA 구현에는 RC카의 상태를 알아야 하며 이를 위해 3축 가속도 센서와 3축 자이로 센서로 이루어진 IMU 센서를 사용한다. 2WD 방식의 주행을 위해 조향에 전륜조향 서보 모터, 구동에 후륜구동 모터를 사용한다. 해당 내용을 고려한 RC카의 부품 구성도(좌)와 부품 간 연결도(우)는 아래와 같다.

도표이(가) 표시된 사진

자동 생성된 설명도표이(가) 표시된 사진

자동 생성된 설명

또한, 부품 간 연결도를 바탕으로 작성한 전체 시스템의 Data Flow는 아래와 같다.

차트, 도표이(가) 표시된 사진

자동 생성된 설명

아두이노에 들어오는 입출력은 다음과 같다.

입력: SW, ACC에 대한 ON/OFF 명령, 카메라 모듈에서 인식된 물체 정보, 초음파 센서에서 수집된 거리 정보, IMU 센서로부터 수집된 RC카의 종/횡방향 가속도, Z축 각속도

출력: LAS, AAS, FAS, 조향 서보 모터에 인가할 전압, 구동 모터에 인가할 전압

클러스터는 GUI를 통해 아래의 내용을 출력한다.

LAS, AAS, FAS, SW의 ON/OFF 스위치, ACC의 ON/OFF 스위치

소프트웨어는 판단SW, 구동SW, 클러스터로 구성되며, 판단SW와 구동SW는 아두이노에 내장되고 클러스터는 PC에서 GUI로 실행된다.

판단SW는 LFA, ACC, FCA의 알고리즘과 DMM(Decision Making Model)으로 구성되어 있으며 주요 기능은 차선 및 물체 인식 및 추적, 주행 상황 판단 등이 있다. 추가로 판단 SW는 각 기능의 상태 정보를 PC로 전송한다.

구동SW는 VCU로 표현하기로 하며, VCU는 모터에 인가할 전압을 계산해 모터를 구동한다.

* 1. 기능 (Statements of purpose)
     1. RC카 전체 제어 – 판단SW의 DMM 참조
     2. 판단SW
        1. LFA

(Info) LFA는 RC Car standby 모드와 RC Car driving 모드에서 기능을 수행한다.

LFA는 PIXY2 카메라 모듈에서 수신한 데이터를 통해 차선의 정보를 확인해야 한다.

LFA는 IMU 센서로부터 수신한 데이터를 통해 차량의 정보를 확인한다.

LFA는 차선 정보, 차량 정보를 이용해 목표 조향각을 계산해야 한다.

LFA는 목표 조향각, 차량 정보, 인식된 차선 개수를 DMM에 송신해야 한다.

* + - 1. ACC

(Info) ACC는 RC Car driving 모드에서 기능을 수행한다.

(Info) 추종 대상 물체와 유지해야 하는 거리 값은 (기본 설정 거리 값)으로 한다.

ACC는 DMM으로부터 ACC ON 명령을 수신 시, 기능을 수행해야 한다.

ACC는 DMM으로부터 ACC ON 명령을 수신 시, DMM으로부터 차량 정보를 수신해야 한다.

ACC는 PIXY2 카메라 모듈에서 수신한 데이터를 통해 추종 대상 물체의 유무를 확인한다.

ACC는 추종 대상 물체가 존재하면 DMM으로부터 수신한 차량 정보와 초음파 센서로부터 수신한 데이터, (기본 설정 거리 값)을 통해 목표 가속도 값을 계산한다.

ACC는 목표 가속도와 추종 대상 물체 여부를 DMM에 송신한다.

* + - 1. FCA

(Info) FCA는 RC Car driving 모드에서 기능을 수행한다.

(Info) FCA 긴급 제동 동작을 수행하기 위한 장애물과의 TTC 값은 (FCA 기본 설정 값)로 한다.

FCA는 PIXY2 카메라 모듈에서 수신한 데이터를 통해 장애물 유무를 확인해야 한다.

FCA는 장애물이 존재하면 DMM으로부터 차량 정보를 수신해야 한다.

FCA는 장애물이 존재하면 IMU 센서로부터 차량 정보를 수신해야 한다.

FCA는 장애물이 존재하면 초음파 센서로부터 수신한 데이터와 DMM으로부터 수신한 차량 정보를 통해 TTC를 계산해야 한다.

FCA는 TTC가 (설정 값) 이내일 경우, 목표 가속도 값을 계산해야 한다.

FCA는 목표 가속도 및 장애물 여부를 DMM으로 송신해야 한다.

* + - 1. DMM

(info) VCU는 Start 모드와 Exit 모드를 제외한 모든 모드에서 기능을 수행한다.

DMM 은 Operating Mode에 따라 RC car를 제어해야 한다.

DMM 은 RC car의 초기화 기능을 수행해야 한다.

DMM은 Operating Mode와 수신한 정보를 이용해 LAS, AAS, FAS를 결정해야 한다.

DMM은 Operating Mode와 수신한 정보를 이용하여 목표 가속도와 목표 조향각을 결정해야 한다.

DMM은 LAS, AAS, FAS를 클러스터로 송신해야 한다.

DMM은 목표 가속도와 목표 조향각을 VCU로 송신해야 한다.

* + 1. 구동SW (VCU)
       1. VCU

(Info) VCU는 모터의 구동을 담당한다.

(info) VCU는 Start 모드와 Exit 모드를 제외한 모든 모드에서 기능을 수행한다.

VCU는 DMM으로부터 수신한 목표 가속도와 목표 조향각을 모터에 인가할 전압으로 계산해야 한다.

* 1. 동작 모드 (Operating Mode)

도표이(가) 표시된 사진

자동 생성된 설명

* 1. 사용자 특징

사용자는 11.1V 4200mah 25C 3S1P LiPo 배터리를 이용해 HW 구동에 필요한 전원을 공급한다.

아두이노에 전원이 공급되면 사용자는 블루투스 모듈에 전원이 공급되는 것을 확인하고 PC의 블루투스 기능을 켠 뒤, 아두이노와 PC 간 블루투스 연결을 시도한다.

아두이노와 PC의 블루투스 연결이 이루어지면 사용자는 PC에서 블루투스 모듈을 통해 ON/OFF 명령을 인가하여 SW를 구동한다.

* 1. 제약 및 가정 사항

DMM은 판단SW가 실행될 때 작동하는 것으로 가정하여, 동작 상태(Active State)는 따로 정의하지 않는다.

* + 1. 주행 환경에 대한 가정

주행 도로는 주행에 장애가 되지 않는 평평한 지면으로 가정한다.

카메라 모듈의 원활한 인식을 위해 조도 500lux (TBD) 이상의 환경이 갖춰져야 한다.

주행 도로는 차선으로 구분되며 직선 구간과 곡선 구간으로 나누며 두 구간 모두 차선의 폭을 0.5m(TBD)로 지정한다.

곡선 구간은 90˚ 코너링 상황을 가정한다.

테스터가 작은 물체를 들고 이동하는 방식을 통해 추종 대상 물체의 주행 상황을 모사한다.

주행 환경에는 차선과 테스트에 필요한 추종 대상 물체 및 장애물을 제외한 물체는 존재하지 않도록 한다.

* + 1. 주행 모델에 대한 가정

현가는 고려하지 않으며 구동, 조향, 제동은 제어기를 적용하지 않는다.

RC카를 단순한 하나의 강체로 취급한다.

RC카의 휠 슬립은 고려하지 않는다.

주행 간 RC카의 Z축 병진 운동, X축 회전 운동에 의한 Rolling, Y축 회전 운동에 의한 Pitching은 고려하지 않는다.

* + 1. RC카 움직임에 대한 제약

RC카의 최대 주행 속력은 1m/s (TBD) 로 지정한다.

RC카 휠의 최대 조향각은 30˚ (TBD) 로 가정한다.

* + 1. 인식 대상 정의

추종 대상 물체는 빨간색(TBD, RGB 범위 구체적 명시)을 인식하도록 한다.

차선은 검은색(TBD, RGB 범위 구체적 명시)을 인식하도록 한다.

PIXY2 카메라 모듈에서 인식된 객체 중 추종 대상 물체와 차선의 색상 범위를 벗어난 것은 모두 장애물로 간주한다.

* + 1. 주행 상황 정의

시나리오 생성을 위해 다음의 다섯 가지 주행 상황을 고려한다.

①전방 물체의 Cut-in

②전방 물체가 도로를 따라 주행

③전방 물체 없이 주행

④전방 장애물 존재

⑤전방 물체의 Cut-out

* 1. HW 사양
     1. Arduino UNO R3 (MCU)
     2. PIXY2 (Camera)
     3. HC-SR04 (Ultra Sensor)
     4. MPU6050 (IMU)
     5. HC-06 (Bluetooth Module)
     6. 11kg MG996R (Servo Motor)
     7. GB37-520 Motor (Driving Motor)
     8. 11.1V 4200mah 25C 3S1P LiPo Battery

1. 세부 요구사항
   1. 외부 인터페이스
      1. 사용자 인터페이스

사용자는 PC 내 클러스터 GUI로부터 판단SW의 ON/OFF 명령을 블루투스 통신으로 아두이노에 인가한다.

* + 1. HW 인터페이스

2.1 절의 시스템 Control Loop를 참고하여 전체 시스템에 대한 HW 인터페이스를 다음과 같이 설명할 수 있다.

PC에서 아두이노 내부 SW를 구동하기 위한 ON/OFF 명령을 블루투스 모듈을 사용한 블루투스 통신을 통해 아두이노에 인가한다.

PIXY2 카메라 모듈에서 인식한 전방의 차선 및 물체에 대한 정보를 I2C 통신을 이용해 아두이노로 송신한다.

초음파 센서에서 전방 물체에 대한 거리 정보를 I2C 통신을 이용해 아두이노로 송신한다.

IMU 센서에서 RC카의 상태 값을 I2C 통신을 이용해 아두이노로 송신한다.

(모터 인터페이스 내용 - 제품 받고나서 작성)

* + 1. 통신 인터페이스

PC에서 SW ON/OFF 명령을 아두이노에 인가하고 아두이노에 수신되는 데이터와 SW 출력 값을 PC에 설치된 아두이노 IDE 내 시리얼 모니터에서 확인하기 위해 PC와 아두이노를 블루투스 통신으로 연결하며 블루투스 모듈을 사용한다.

* + 1. 논리적 인터페이스

PIXY2 → Arduino

- 물체 정보 (x 위치, y위치, 색, )

- 라인 정보

Arduino

* 1. 기능 요구사항
     1. LFA

LFA는 PIXY2 카메라 모듈에서 수신한 데이터를 통해 차선의 정보를 확인해야 한다.

* LFA는 RC car init 모드에서 RC car standby 모드로 전이되면, PIXY2 카메라 모듈로부터 차선 색상 범위에 해당하는 차선의 개수를 계산해야 한다.

LFA는 IMU 센서로부터 수신한 데이터를 통해 차량의 정보를 확인한다.

* LFA는 RC car init 모드에서 RC car standby 모드로 전이되면, 인식된 차선의 개수와 관계없이 IMU 센서로부터 차량의 종/횡방향 가속도와 Z축 각속도를 수신해야 한다.
* LFA는 IMU 센서로부터 수신한 차량의 종/횡방향 가속도를 통해 차량의 종/횡방향 속도를 계산해야 한다.
* LFA는 IMU 센서로부터 수신한 차량의 Z축 각속도를 통해 차량의 Z축 자세각을 계산해야 한다.

LFA는 차선 정보, 차량 정보를 이용해 목표 조향각을 계산해야 한다.

* LFA는 RC car standby 모드 혹은 RC car driving 모드에서 인식된 차선 개수가 1개 이상일 경우, (기본 주행 값)을 알고리즘에 적용해 차선 정보와 차량 정보를 이용해 목표 조향각을 계산해야 한다.
* LFA는 RC car driving 모드에서 DMM으로부터 ACC의 목표 가속도 값을 수신할 경우, 이를 LFA 알고리즘에 적용하여 목표 조향각을 계산해야 한다.
* LFA는 RC car driving 모드에서 DMM으로부터 ACC의 목표 가속도 값을 수신하지 않을 경우, (기본 주행 값)을 알고리즘에 적용하여 목표 조향각을 계산해야 한다.

LFA는 목표 조향각, 차량 정보, 인식된 차선 개수를 DMM에 송신해야 한다.

* LFA는 인식된 차선의 개수가 0개일 경우, 차량의 종/횡방향 속도, 차량의 Z축 자세 각, 인식된 차선의 개수를 DMM에 송신해야 한다.
* LFA는 인식된 차선의 개수가 1개 이상일 경우, 목표 조향각, 차량의 종/횡방향 속도, 차량의 Z축 자세 각, 인식된 차선의 개수를 DMM에 송신해야 한다.
  + 1. ACC

ACC는 DMM으로부터 ACC ON 명령을 수신하면 기능을 수행해야 한다.

* ACC는 RC car driving 모드에서 DMM으로부터 ACC ON 명령 수신 시, 기능을 수행해야 한다.

ACC는 DMM으로부터 ACC ON 명령 수신 시, DMM으로부터 차량 정보를 수신해야 한다.

* ACC는 RC car driving 모드에서 DMM으로부터 ACC ON 명령 수신 시, LFA로부터 차량의 종/횡방향 속도 값을 수신해야 한다.

ACC는 PIXY2 카메라 모듈에서 수신한 데이터를 통해 추종 대상 물체의 유무를 확인해야 한다.

* ACC는 RC car driving 모드에서 DMM으로부터 ACC 명령을 수신한 뒤, 차량 전방에 PIXY2 카메라 모듈을 통해 물체를 인식하면, 물체의 색상 범위를 통해 추종 대상 물체 여부를 판단해야 한다.

ACC는 추종 대상 물체가 존재하면, DMM으로부터 수신한 차량 정보와 초음파 센서로부터 수신한 데이터, (기본 설정 거리 값)을 통해 목표 가속도 값을 계산해야 한다.

* ACC는 RC car driving 모드에서 DMM으로부터 ACC 명령을 수신한 뒤, 추종 대상 물체가 존재하면, 초음파 센서 3개로부터 수신한 데이터와 DMM으로부터 수신한 차량 정보를 알고리즘에 적용해 추종 대상 물체의 상대 거리 값 및 추종 대상 물체의 상대 속도 값을 계산해야 한다.
* ACC는 추종 대상 물체의 상대 거리 값과 추종 대상 물체의 상대 속도 값, (기본 설정 거리 값)을 이용해 1초 이내에 (기본 설정 거리 값)과 상대 속도 0에 도달하는 목표 가속도 값을 계산해야 한다.

ACC는 목표 가속도와 추종 대상 물체 여부를 DMM에 송신해야 한다.

* ACC는 RC car driving 모드에서 DMM으로부터 ACC 명령을 수신한 뒤, 추종 대상 물체가 존재하면 목표 가속도와 추종 대상 물체 여부를 DMM에 송신해야 한다.
* ACC는 RC car driving 모드에서 DMM으로부터 ACC 명령을 수신한 뒤, 추종 대상 물체가 존재하지 않으면 추종 대상 물체 여부를 DMM에 송신해야 한다.
  + 1. FCA

FCA는 PIXY2 카메라 모듈에서 수신한 데이터를 통해 장애물 유무를 확인하다.

* FCA는 RC car driving 모드에서 차량 전방에 PIXY2 카메라 모듈을 통해 인식된 물체가 있을 때, 물체의 색상 범위를 통해 장애물 여부를 판단해야 한다.

FCA는 장애물이 존재하면 DMM으로부터 차량 정보를 수신해야 한다.

* FCA는 RC car driving 모드에서 장애물이 존재하면, DMM으로부터 LFA에서 계산된 차량의 종/횡방향 속도 값을 수신해야 한다.

FCA는 장애물이 존재하면 IMU 센서로부터 차량 정보를 수신해야 한다.

* FCA는 RC car driving 모드에서 장애물이 존재하면, IMU 센서로부터 차량의 종/횡방향 가속도 값을 수신해야 한다.

FCA는 장애물이 존재하면 초음파 센서로부터 수신한 데이터와 DMM으로부터 수신한 차량 정보를 통해 TTC 값을 계산해야 한다.

* FCA는 RC car driving 모드에서 장애물이 존재하면, 초음파 센서에서 수신한 데이터를 통해 장애물과의 거리 값을 계산해야 한다.
* FCA는 장애물과의 거리 값, 차량의 종/횡방향 속도 및 가속도 값을 이용해 TTC 값을 계산해야 한다.

FCA는 TTC가 (설정 값) 이내일 경우, 목표 가속도 값을 계산해야 한다.

* FCA는 RC car driving 모드에서 장애물과의 TTC가 (설정 값) 이내일 경우, 장애물과의 거리 값, 차량의 종/횡방향 속도 및 가속도 값을 이용해 TTC 보다 작은 시간 내 차량이 정지하도록 긴급 제동 가속도를 계산한다. 그렇지 않을 경우 긴급 제동 가속도는 0으로 한다.

FCA는 목표 가속도 및 장애물 여부를 DMM으로 송신해야 한다.

* FCA는 RC car driving 모드에서 장애물이 존재하고 긴급 제동 가속도가 0이 아닐 경우, 긴급 제동 가속도와 장애물 여부를 DMM으로 송신해야 한다.
* FCA는 RC car driving 모드에서 장애물이 존재하고 긴급 제동 가속도가 0일 경우, 장애물 여부를 DMM으로 송신해야 한다.
  + 1. DMM

DMM은 Operating Mode에 따라 RC car를 제어해야 한다.

* DMM은 모든 모드에서 클러스터와 Connection 여부를 주기적으로 확인해야 한다.
* DMM은 System standby 모드를 제외한 모든 모드에서 클러스터와 Disconnection되면 System standby 모드에 대한 reset을 수행해야 한다.
* DMM은 System standby 모드에서 클러스터와 Connection되었을 때, 주기적으로 SW ON/OFF명령을 확인해야 한다.
* DMM은 System standby 모드에서 클러스터와 Connection되었을 때, SW ON 명령 수신 시 RC Car init 모드로 전이해야 한다.
* DMM은 System standby 모드에서 클러스터와 Connection되었을 때, SW OFF 명령 수신 시 RC car를 종료해야 한다.
* DMM은 System standby 모드에서 클러스터와 Connection되었을 때, ACC ON/OFF 명령 수신(input fault)시 System standby 모드에 대한 reset을 수행해야 한다.
* DMM은 RC car init 모드에서 fault 발생 시 RC car init 모드에 대한 reset을 수행해야 한다.
* DMM은 RC car init 모드, RC car driving 모드에서 클러스터로부터 SW OFF 명령 수신 시 RC car를 종료해야 한다.
* DMM은 System standby 모드, RC car init 모드에서 각 모드에 대한 reset을 4회 이상 수행 시, RC car를 종료해야 한다.
* DMM은 RC car standby 모드에서 RC car standby에 대한 reset을 11회 이상 수행 시, RC car를 종료해야 한다.
* DMM은 RC car standby 모드, RC car driving 모드에서 인식된 차선 개수가 0개가 되면, RC Car Standby 모드에 대한 reset을 수행해야 한다.
* DMM은 RC car standby 모드에서 인식된 차선 개수가 1개 이상이면, RC car driving 모드로 전이해야 한다.

DMM은 RC car의 초기화 기능을 수행해야 한다.

* DMM은 RC Car init 모드에서 DMM 은 Camera HW를 초기화해야 한다.
* DMM은 RC Car init 모드에서 DMM 은 Ultrasonic Sensor HW를 초기화해야 한다.
* DMM은 RC Car init 모드에서 DMM 은 IMU HW를 초기화해야 한다.
* DMM은 RC Car init 모드에서 DMM 은 LAS, AAS, FAS를 OFF 로 설정해야 한다.

DMM은 Operating Mode와 수신한 정보를 이용해 LAS, AAS, FAS를 결정해야 한다.

* DMM은 System standby 모드를 제외한 모든 모드에서 클러스터와 Disconnection되면 LAS, AAS, FAS를 OFF로 변경해야 한다.
* DMM은 RC car standby 모드, RC car driving 모드를 제외한 모든 모드에서 LAS, AAS, FAS를 OFF로 변경해야 한다.
* DMM은 RC car init 모드에서 RC car standby 모드로 전이하면 LAS를 ON으로 변경해야 한다.
* DMM은 RC car driving 모드에서 FCA로부터 수신한 목표 가속도가 0이 아닐 경우, FAS를 ON으로 (설정한 시간)동안 설정한 뒤 다시 OFF로 변경해야 한다.
* DMM은 RC car driving 모드에서 FCA로부터 수신한 목표 가속도가 0일 경우, FAS를 OFF로 변경해야 한다.
* DMM은 RC car standby 모드, RC car driving 모드에서 PIXY2 카메라 모듈이 정상 작동할 경우 LAS를 ON으로 변경해야 한다.
* DMM은 RC car driving 모드에서 클러스터로부터 ACC ON 명령 수신 시, AAS를 ON으로 변경해야 한다.
* DMM은 RC car driving 모드에서 클러스터로부터 ACC OFF 명령 수신 시, AAS를 OFF로 변경해야 한다.
* DMM은 RC Car Driving 모드에서 인식된 차선 개수가 0개가 되면, AAS, FAS를 OFF로 변경해야 한다.

DMM은 Operating Mode와 수신한 정보를 이용하여 목표 가속도와 목표 조향각을 결정해야 한다.

* DMM은 RC car driving 모드에서 FAS가 ON인 경우, AAS의 ON/OFF 여부와 상관없이 FCA의 목표 가속도와 LFA의 목표 조향각을 선정해야 한다.
* DMM은 RC car driving 모드에서 FAS가 OFF, AAS가 ON인 경우, ACC의 추종 대상 물체 여부에 따른 목표 가속도와 LFA의 목표 조향각을 선정해야 한다.
* DMM은 RC car driving 모드에서 FAS가 OFF, AAS가 OFF일 경우, LFA의 목표 가속도와 목표 조향각을 선정해야 한다.
* DMM은 RC Car Driving 모드에서 다음의 4가지 경우에 를 계산하여 목표 가속도로 선정해야 한다. 이때, 목표 조향각은 LFA의 목표 조향각을 선정해야 한다.

1. Disconnection 될 경우
2. 클러스터로부터 SW OFF 명령 수신 시
3. 인식된 차선의 개수가 0개일 경우
4. sensor나 actuator fault가 발생할 경우
   * : 1초 이내 RC Car를 정지할 수 있는 가속도

DMM은 LAS, AAS, FAS를 클러스터로 송신해야 한다.

* DMM은 시스템 시작 및 종료를 제외한 모든 모드에서 LAS, AAS, FAS를 클러스터로 송신해야 한다.
  + 1. VCU

VCU는 DMM으로부터 수신한 목표 가속도와 목표 조향각을 모터에 인가할 전압으로 계산해야 한다.

* VCU는 DMM으로부터 수신한 목표 가속도를 구동 모터에 인가할 전압으로 계산해야 한다.
* VCU는 DMM으로부터 수신한 목표 조향각을 조향 서보 모터에 인가할 전압으로 계산해야 한다.

(모터 전압 계산에 대한 요구사항이 있어야 함)