

# 대학생 자작자동차대회 자율주행 포물러 경진대회 규정

[ 시행 2025.7.24. 이사회 ]

제1장. 개요 .....	2
제1조. 목적 .....	2
제2조. 적용대상 .....	2
제3조. 용어 정의 .....	2
제4조. 대회 관계자의 권한 및 참가 팀 행동 규범 .....	2
제5조. 참가 자격 및 팀 구성 .....	2
제6조. 규정 개정 .....	3
제2장. 차량 및 시뮬레이터 기술 규정 .....	3
제7조. 시뮬레이터 .....	3
제1항. 차량 모델 .....	3
제2항. 센서 .....	4
제8조. PC 및 개발 환경 제한 .....	5
3. 자율주행 알고리즘 개발규정 .....	6
제9조. 개발 범위 .....	6
제10조. 자율주행 시스템 상태(AS STATUS) 정의 .....	6
제11조. ROS 통합규정 .....	6
4. 미션 규정 .....	7
제12조. TRACK DRIVE 미션 .....	7
제1항. 운영 구성 .....	7
제2항. 트랙 구성 .....	7
제13조. 패널티 및 실격 규정 .....	7
5. 운영규정 .....	7
제14조. 총점 및 최종 순위 결정 .....	7
제15조. 랩 타임 측정 .....	8
제16조. 제출물 .....	8
제1항. 개발보고서 .....	8
제2항. 알고리즘 .....	8
제17조. 가상 원격 긴급 정지 시스템 (VIRTUAL RES) .....	8
제18조. 대회 중 기술적 제한 .....	8
제19조. 예선(QUALIFYING) .....	8
제20조. 본선(FINALS) .....	9
6. 차년도 포물러 자율주행 경진대회 대비 규정 .....	9

## 제1장 개요

### 제1조 (목적)

본 규정은 2025 자율주행 포뮬러 경진대회 경기진행에 관한 사항을 규정함을 목적으로 한다.

### 제2조 (적용대상)

본 규정의 적용 대상은 2025 자율주행 포뮬러 경진대회에 참여 신청을 제출하고 승인된 모든 참가 팀 및 경진대회 준비, 운영에 참여하는 모든 인원으로 한다.

### 제3조 (용어 정의)

본 규정에서 사용하는 용어의 정의는 다음과 같다.

- ① 자율주행 차량: 인간의 운전 없이 자동으로 주행할 수 있는 자동차로, 대회에서 규정되거나 허용된 차량에 장착된 센서 등을 활용하여, 제어기에 의해 목적한 경로를 주행하며, 대회에서 주어진 미션을 수행하는 차량으로 작동 중 인간의 수동 개입이 필요하지 않은 차량을 의미한다.
- ② 시뮬레이션 컴퓨터: 자율주행 차량 및 자율주행 차량에 장착된 센서의 출력을 생성하여 참가자 컴퓨터에 통신을 통해 전달하고, 참가자 컴퓨터로부터 제어 명령을 입력 받으며, 자율주행 차량의 입출력 정보를 이용하여 차량 주행 및 주행 환경을 시뮬레이션에서 수행하는 컴퓨터를 의미한다.
- ③ 알고리즘 컴퓨터: 자율주행 차량에 각종 센서의 출력을 받아, 인지, 판단, 제어 등에 필요한 연산을 수행하고, 자율주행 차량에 제어 명령을 출력하는 컴퓨터를 의미한다.
- ④ 자율주행 시스템 책임자(ASR, Autonomous System Responsible): 각 팀에서 자율주행 시스템의 운영과 안전을 총괄하는 책임자로 지정된 팀원이다.
- ⑤ 오퍼레이터: 대회 진행 시 운영하는 팀 또는 프로그램으로, 가상 원격 긴급 정지 시스템(Virtual RES)을 통해 차량에 출발 신호 전달 및 긴급 정지 명령을 내리고 탈락 여부를 모니터링하는 역할을 한다.
- ⑥ ROS: Robot Operating System의 약자로 본 대회에서는 ROS1을 의미한다.
- ⑦ FSDS: Formula Student Driverless Simulator의 약자로 본 대회에서 사용하는 시뮬레이션을 의미한다.
- ⑧ DNF: Did Not Finish의 약자로 주행을 시작하였으나 완주하지 못한 경우를 말한다.
- ⑨ DQ: Disqualified의 약자로 규칙 위반 등으로 인해 해당 경기의 점수를 획득할 자격을 잃는 경우를 의미한다.
- ⑩ 위원회: 자율주행 포뮬러 경진대회를 운영 및 심사 평가 위원들을 칭한다.
- ⑪ 참가팀: 자율주행 포뮬러 경진대회에 참가 신청서를 제출하고 참가비를 제출한 팀을 의미한다.

### 제4조 (대회 관계자의 권한 및 참가 팀 행동 규범)

- ① 운영진은 대회의 안전하고 효율적인 운영을 위해 필요하다고 판단될 경우, 언제든지 대회 일정을 변경하거나 규칙을 해석 및 수정할 권한을 가진다.
- ② 참가팀은 비신사적인 행위를 할 경우 페널티를 받을 수 있으며, 반복될 경우 대회에서 실격될 수 있다.
- ③ 규칙에 대한 질문은 지정된 절차에 따라 문의해야 한다.
- ④ 점수, 규칙 해석 등 공식적인 결정에 이의가 있는 팀은 팀 대표를 통해 정해진 절차에 따라 이의를 제기할 수 있다.

### 제5조 (참가 자격 및 팀 구성)

- ① 참가 팀장과 팀원은 대회 기간에 학부에 재학 중인 학생이어야 한다.

- ② 각 팀은 반드시 1명의 자율주행 시스템 책임자(ASR, Autonomous System Responsible)를 지정해야 한다. ASR은 팀의 자율주행 시스템 운영을 총괄한다.
- ③ 국내·외 모든 대학(교)의 학부생으로서 학회 학생 회원이어야 하며, 동일 대학(교) 학부생 간 또는 타 대학(교) 학부생 간 연합으로 참가 팀 구성이 가능하다.
- ④ 참가 팀 인원은 지도교수 1명, 팀장 1명을 포함하여 최대 10명까지 등록 가능하다.
- ⑤ 지도교수는 학회 정회원 또는 종신회원이어야 한다.

#### 제6조 (규정 개정)

본 규정의 내용은 대회 운영진의 판단에 따라 변경될 수 있다.

## 제2장 차량 및 시뮬레이터 기술 규정

#### 제7조 (시뮬레이터)

- Engine: Unreal Engine 4.27 기반의 Formula Student Driverless Simulator (버전 2.2.0)
- 운영체제 환경: Ubuntu 20.04.
- 다운로드: 대회에서 사용될 시뮬레이터는 안내된 설치 경로를 통해 다운로드 받아야 한다.  
ROS Bridge: <https://github.com/FS-Driverless/Formula-Student-Driverless-Simulator>  
Simulator: <https://github.com/FS-Driverless/Formula-Student-Driverless-Simulator/releases>
- 시뮬레이터 document 링크: <https://fs-driverless.github.io/Formula-Student-Driverless-Simulator/v2.2.0/>
- 상기 버전은 상황에 따라 업데이트되거나 변경될 수 있다.

#### ① 차량 모델



그림1. 시뮬레이션 차량 모델

1. Model: TechnionCar (FSOnline 2020 competition).
2. Exterior Dimensions: Length: 1.8 m, Width: 1.0 m, Height: 0.5 m, Wheelbase: 1.55
3. Physical Parameters: Mass: 255 kg, Max speed: 27 m/s, Drag coefficient: 0.3
4. Vehicle Center: 차량의 센터(car pawn center)는 무게중심 아래 지면에 위치한다. 다른 센서들은 이 위치를 기준으로 위치한다.

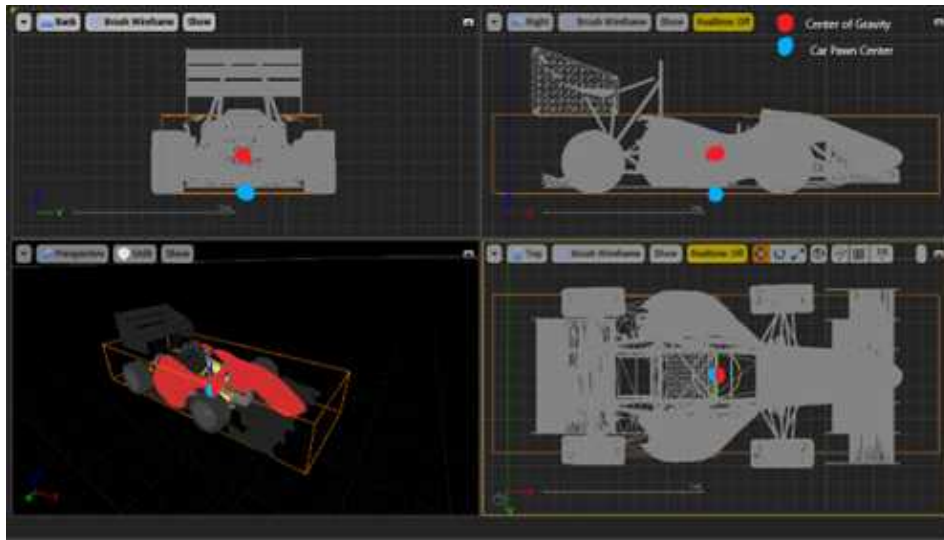


그림2. 시뮬레이션 차량 구성

② 센서

- 대회 중 시뮬레이션은 아래와 같은 센서 설정을 가지며 참가자들은 아래 센서 구성을 이용해 자율주행을 수행하여야 한다.
- 센서 데이터는 무작위 노이즈가 추가될 수 있다.

1. 카메라

가. 카메라의 자동 노출, 모션 블러, 감마 설정은 모든 팀에 동일하게 적용된다.

나. 사양 (Left Camera)

- a. Type: RGB 카메라
- b. Field of View: 90°
- c. Width × Height: 640 × 480
- d. Position (x, y, z): -0.43, 0.15, 0.82 (m)
- e. Rotation (roll, pitch, yaw): 0.0, 0.0, 0.0 (deg)
- f. Frame rate: 30 Hz

다. 사양 (Right Camera)

- a. Type: RGB 카메라
- b. Field of View: 90°
- c. Width × Height: 640 × 480
- d. Position (x, y, z): -0.43, -0.15, 0.82 (m)
- e. Rotation (roll, pitch, yaw): 0.0, 0.0, 0.0 (deg)
- f. Frame rate: 30 Hz

라. 사용 가능 ROS토픽: (좌) /fdds/cameracam1, (우) /fdds/cameracam2

2. LiDAR

가. 위치: 차량 Center 최하단 기준.

나. 사양

- a. Channels: 16
- b. Position (x, y, z): 0.60, 0.0, 0.10 (m)
- c. Rotation (roll, pitch, yaw): 0.0, 0.0, 0.0 (deg)
- d. PointsPerScan: ~~1800~~3600
- e. Vertical FoV: ~~-15, -15~~ -5, 5

- e. Horizontal FoV: -180, 180
  - f. Rotation per Second: 10 Hz
  - g. 사용 가능 ROS토픽: /fsds/lidar/Lidar1
3. GPS
- 가. 사양
    - a. Position (x, y, z): 0, 0, 0
    - b. Frame rate: 10 Hz
    - c. 사용 가능 토픽: /fsds/gps
4. IMU
- 가. 사양
    - a. Position (x, y, z): 0, 0, 0
    - b. Frame rate: 250 Hz
    - c. 사용 가능 토픽: /fsds/imu
5. 액추에이터
- 가. 스티어링 휠
    - a. 범위: [-1, 1]
  - 나. 가속 페달
    - a. 범위: [0, 1]
  - 다. 브레이크 페달
    - a. 범위: [0, 1]
  - 라. 사용가능 토픽 : /fsds/control\_command

제8조 (PC 및 개발 환경 제한)

- ① 대회 중 참가자는 준비된 알고리즘 컴퓨터만 사용이 허용된다.
- ② 시뮬레이터는 시뮬레이터 컴퓨터에서 동작하며 참가팀의 자율주행 알고리즘은 알고리즘 컴퓨터에서 구동된다.

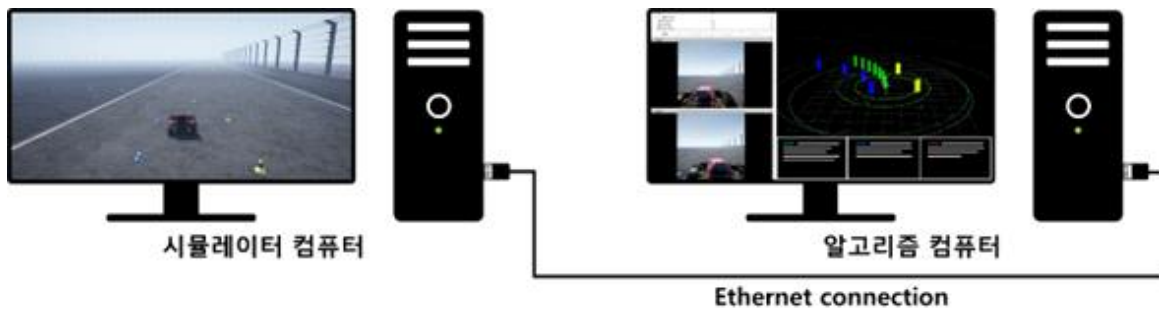


그림3. 시뮬레이터 컴퓨터 및 알고리즘 컴퓨터 운영 환경

- ③ 시뮬레이터 컴퓨터와 알고리즘 컴퓨터는 LAN으로 연결되며 통신한다.
- ④ OS는 Ubuntu 20.04 만 사용한다.
- ⑤ ROS 버전은 Noetic을 사용한다.
- ⑥ 시뮬레이터 및 알고리즘 컴퓨터의 성능은 아래와 같은 수준이다.
  - 1. CPU: Intel I7 14th 급
  - 2. GPU: Nvidia RTX5070 급
  - 3. RAM: DDR5 64GB 급

### 제3장 자율주행 알고리즘 개발 규정

#### 제9조 (개발 범위)

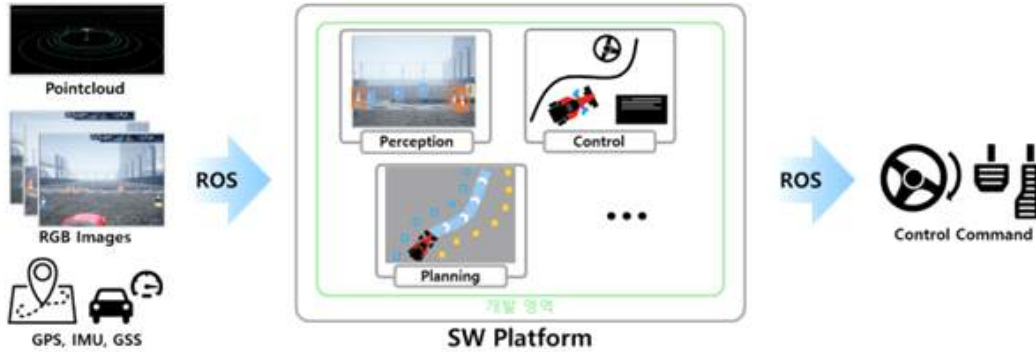


그림4. 소프트웨어 플랫폼 운영 환경

- ① 참가팀은 아래의 입출력 데이터를 활용하여 자율주행 알고리즘을 개발해야 한다.
  1. 입력 데이터: LiDAR PointCloud(`sensor_msgs/PointCloud2`), RGB Camera Images(`sensor_msgs/Image`), IMU(`sensor_msgs/Imu`), GPS(`sensor_msgs/NavSatFix`).
  2. 출발 및 긴급 정지 신호: Go Signal, Emergency Stop
  3. 출력 데이터: Control Command (`fs_msgs/ControlCommand`).
- ② 개발 환경: 제공된 SW 플랫폼 내에서 ROS를 통해 시뮬레이터로부터 센서 데이터를 받고, ROS로 제어 명령(Control Command)을 출력해야 한다.
- ③ 제공된 SW 플랫폼에서 수정가능한 파일은 아래와 같다.
  - Cpp이용시: `formula_autonomous_system.hpp`, `formula_autonomous_system.cpp`
  - Python 이용시: `formula_autonomous_system.py`

#### 제9조 -1 (입력 데이터)

##### LiDAR Point Cloud

- 타입: `sensor_msgs/PointCloud2`
- 가상의 LiDAR 센서로 주변 환경을 레이저로 측정한 Point Cloud 정보가 제공된다.

##### RGB Camera Images

- 타입: `sensor_msgs/Image`
- 가상의 Camera 센서로 RGB 이미지 정보가 제공된다.

##### IMU (Inertial Measurement Unit)

- 타입: `sensor_msgs/Imu`
- 가상의 IMU 센서로 차량의 가속도와 각속도, 방향 정보가 제공된다.

##### GPS (Global Positioning System)

- 타입: `sensor_msgs/NavSatFix`
- 가상의 GPS 센서로 차량의 위치정보가 지리 좌표계(latitude, longitude, altitude)로 제공된다.

#### 제9조 -2 (출발 및 긴급 정지 신호)

## Go Signal

-타입: fs\_msgs/GoSignal

-미션과 트랙에 대한 정보가 문자열로 제공된다.

-자율주행차량은 오퍼레이터로부터 해당 메시지를 받은 이후 출발할 수 있다.

## 제9조 -3 (출력 데이터)

### Control Command

-타입: fs\_msgs/ControlCommand

-참가팀의 자율주행 차량은 가속 (throttle), 조향각 (steering), 제동 (brake) 정보를 포함한 Control Command를 통해 차량을 제어할 수 있다.

### AS State

-타입: std\_msgs/String

-자율주행차량 시스템의 상태를 문자열 형태로 "AS Off", "AS Ready", "AS Driving", "AS Emergency", "AS Finished" 중 1개로 1 Hz 이상으로 전송하여야 한다.

## 제10조 (자율주행 시스템 상태(AS Status) 정의)

① 차량의 현재 상태를 명확히 하기 위해 아래와 같은 상태를 정의하며, 이 상태는 시뮬레이터 상에 시각적으로 표시될 수 있다.

1. AS Off: 자율주행 시스템이 비활성화된 상태.
2. AS Ready: 주행 준비가 완료되어 오퍼레이터의 GO 신호를 대기하는 상태.
3. AS Driving: 자율주행으로 트랙을 주행 중인 상태.
4. AS Emergency: 긴급 상황으로 인해 비상 정지 절차가 발동된 상태.
5. AS Finished: 미션을 완료하고 정지한 상태.

② 차량의 상태를 확인하기 위해 참가자는 아래의 토픽을 1Hz 이상의 주기로 출력하여야 한다.

1. 토픽: /fsds/AS\_status

## 제11조 (ROS 통합 규정)

카테고리	토픽명	타입
자율주행차량이 대회 중 수신 가능한 토픽	/fsds/gps	sensor_msgs/NavSatFix
	/fsds/imu	sensor_msgs/Imu
	/fsds/cameracam1	sensor_msgs/Image
	/fsds/cameracam2	sensor_msgs/Image
	/fsds/lidar/Lidar1	sensor_msgs/PointCloud2
측면, 전방 충격 보호 구조, 롤 후프 지지대	/fsds/testing_only/odom	nav_msgs/Odometry
	/fsds/testing_only/track	fs_msgs/Track
오퍼레이터가 발행하는 출발 신호 토픽	/fsds/signal/go	fs_msgs/GoSignal
자율주행차량이 발행하는 종료 신호 토픽	/fsds/signal/finished	fs_msgs/FinishedSignal

카테고리	토픽명	타입
차량 제어 명령 발행 토픽	/fsds/control_command	fs_msgs/ControlCommand
차량 상태 신호 발행 토픽	/fsds/AS_status	std_msgs/String

## 제4장 미션 규정

### 제12조 (Track Drive 미션)

#### ① 운영 구성

1. 트랙 공개: 트랙은 대회 시작 전 공개되며 자율주행 차량은 트랙 정보를 주행전에 제공받지 못한다.

1. 예선: 지정된 트랙을 단독으로 10바퀴 주행하여 시간을 측정한다.

2. 본선: 예선보다 상향된 난이도의 트랙에서 단독으로 주행하여 시간을 측정한다.

3. 출발: 차량의 전륜 바퀴를 시작선 6m 전 공간에 배치하여 출발한다.

4. 정지: 결승선 통과 후 30m 이내의 트랙 내부 최소 한 바퀴 이상 포함된 상태로 정지해야 한다.

#### ② 트랙 구성

1. 최소 도로 폭: TBD 3.0m

2. 콘 간격: TBD

3. 트랙 길이: TBD

4. 트랙 최소 곡률 반경: TBD 9.0m

가. 곡률 반경은 경로 중앙을 기준으로 측정된다.

5. 시작선에는 오렌지 콘이 좌우로 배치된다.

6. 트랙 좌측에는 파란색 콘이 배치되고 우측에는 노란색 콘이 배치된다.

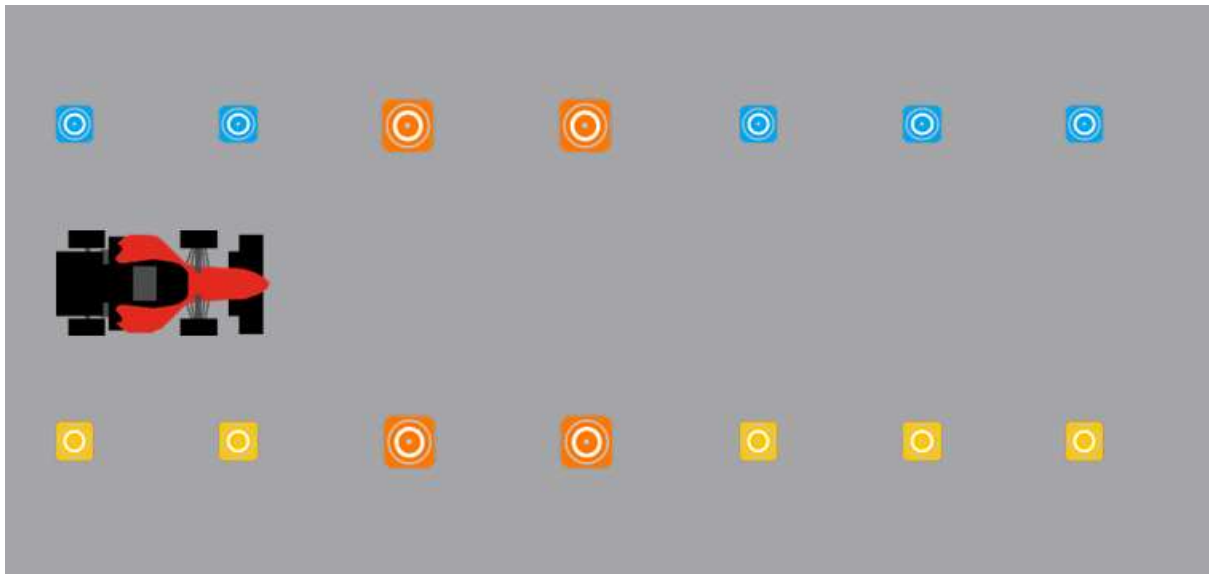


그림5. 트랙 콘 배치 예시

### 제13조 (페널티 및 실격 규정)

페널티 이름	페널티 내용	페널티
Down or Out (DOO)	차량이 콘(Cone)과 충돌하거나 닿을 경우 대회 페널티가 부여된다. 단, 한번 충돌한 콘은 두 번 이상 페널티가 부여되지 않는다.	콘 터치 당 -20점 또는 주행거리 -20m
Off-Course (OC)	차량의 네 바퀴 모두 지정된 경로를 완전히 이탈할 경우, 해당 주행은 중지된다.	실격-DNF
Unsafe Stop (USS)	지정되지 않은 곳에서 정지하거나, 미션을 완료한 후 30초 내에 정지하지 않을 경우 페널티가 부여된다.	-50점
Max Driving Time (MDT)	차량은 출발 후 도착까지 지정된 제한 시간을 준수해야 한다.	실격-DNF
Reckless or Aggressive Driving	심사위원이 비정상적인 자율주행(예: 역주행)으로 판단할 경우 경기를 중지시킬 수 있다.	실격-DNF
Do Not Start (DNS)	오퍼레이터로부터 시작 신호(GO signal)를 받은 후 30초 이내에 출발하지 못할 경우 실격 처리된다.	실격-DNF
Technical Infringement	대회 시작 후 사전에 제출된 알고리즘과 다른 로직을 사용하거나 파라미터를 임의로 수정하는 등 부정행위가 적발될 경우, 즉시 실격 처리된다. 자율주행차량이 사용 불가능한 토픽을 수신하는 경우 실격처리 된다.	실격DQ

## 제5장 운영 규정

### 제??조 (미션 시작)

① 미션 시작을 위해 아래와 같은 절차를 따른다.

1. 위원회는 참가팀으로부터 제출 받은 알고리즘을 빌드한다.
2. 위원회는 지정된 이름의 launch 파일을 실행하여 참가팀의 자율주행차량 알고리즘을 실행시킨다.

### 제14조 (총점 및 최종 순위 결정)

- ① 총점: 설계 보고서 점수 + 예선 Track Drive 점수 + 본선 Track Drive 점수를 합산하여 1,000점 만점으로 최종 순위를 결정한다.
- ② 동점자 처리: 최종 합계 점수가 동일할 경우, 아래 순서의 성적을 비교하여 상위 순위를 결정한다.
  1. 본선 Track Drive 기록
  2. 예선 Track Drive 기록
  3. 설계 보고서 평가 점수

### 제15조 (랩 타임 측정)

- ① 랩 타임은 시뮬레이션 컴퓨터에서 측정된다.
- ② 측정 시작: 오퍼레이터가 출발 신호를 보낸 후 참가자 차량이 출발 선을 통과한 시점부터 랩 타임 측정이 시작된다.
- ③ 랩 기록: 매 랩 타임은 출발선에 차량의 전면부가 닿을 시 기록된다.
- ④ 종료 신호: 각 참가팀은 주어진 미션을 모두 완료한 후 시뮬레이션 컴퓨터에 종료 신호를 보내야 한다.
  1. 토픽명: /fsds/signal/finished

### 제16조 (제출물)

- ① 개발보고서
  1. 모든 참가팀은 대회 전 공지된 기간에 개발 보고서를 제출해야 한다.
  2. 제출 마감일을 준수해야 하며, 지연 제출 시 감점 페널티가 부여될 수 있다.

## ② 알고리즘

1. 모든 참가팀은 대회 시작 전 공지된 기간에 배포된 SW 플랫폼에 기반한 자율주행 알고리즘 파일을 제출해야 한다.
2. 제출 마감일을 준수해야 하며, 지연 제출 시 감점 페널티가 부여될 수 있다.
3. 배포된 SW플랫폼에서 제출 가능한 파일은 cpp 이용시 `formula_autonomous_system.hpp`, `formula_autonomous_system.cpp` 파일을 제출 가능하며 python 이용시 `formula_autonomous_system.py` 파일을 제출 가능하다.

## 제17조 (가상 원격 긴급 정지 시스템 (Virtual RES))

- ① 오퍼레이터는 Virtual RES를 통해 모든 차량의 긴급 정지(EMERGENCY STOP)를 제어한다. 참가팀은 이 신호에 즉시 반응해야 한다.

1. 토픽명: `/fsds/emergency_stop (std_msgs/Bool)`

## 제18조 (대회 중 기술적 제한)

- ① 대회 당일시 알고리즘 컴퓨터의 인터넷 사용은 불가하다.
- ② 자율주행 중 키보드 및 마우스 또는 인터넷 등을 이용한 외부 주행 개입은 절대 불가하다.
- ③ 대회 시작 전 제출된 알고리즘의 수정 및 변경은 불가하다.
- ④ 알고리즘의 구동은 위원회에서 담당한다.

## 제19조 (예선(Qualifying))

- ① 진행: 참가팀은 단독으로 Track Drive를 수행하여 랩 타임을 측정한다.
- ② 예선 성적 상위 N (TBD)개의 팀이 본선에 진출한다.
- ③ 개발 보고서 평가 (150, TBD점): 제출된 보고서는 아래의 항목들을 기준으로 평가된다.
  1. 자율주행 소프트웨어 아키텍처 및 구현 (Software)
  2. 자율주행 기능 구현 및 독창성 (Autonomous Functionality)
  3. 보고서의 완성도 및 명료성 (Engineering Design Report)
- ④ 예선 Track Drive 점수 (300, TBD점): 최고 기록과의 상대 평가로 점수를 산정한다. 완주 시 최소 30점의 완주 점수가 부여된다.
  1. 예선 점수 =  $270 * ((T_{max} / T_{your}) - 1) / ((T_{max} / T_{min}) - 1) + 30$
  2.  $T_{your}$ : 해당 팀의 예선 기록 (페널티 포함)
  3.  $T_{min}$ : 예선 참가팀 중 가장 빠른 기록
  4.  $T_{max}$ :  $T_{min}$ 의 1.45배.  $T_{your}$ 가  $T_{max}$ 를 초과할 경우  $T_{max}$ 로 간주.
- ⑤ 완주하지 못한 경우 주행한 거리를 기반으로 평가된다.

## 제20조 (본선(Finals))

- ① 진행: 예선 상위 N (TBD)개의 팀을 대상으로 진행되며, 예선보다 높은 난이도의 트랙에서 단독 주행으로 최종 시간을 측정한다.
- ② 본선 Track Drive 점수 (550점): 본선 기록을 바탕으로 상대 평가하여 최종 점수를 산정한다. 완주 시 최소 40점의 완주 점수가 부여된다.
  1. 본선 점수 =  $510 * ((T_{max} / T_{your}) - 1) / ((T_{max} / T_{min}) - 1) + 40$
  2.  $T_{your}$ : 해당 팀의 본선 기록 (페널티 포함)
  3.  $T_{min}$ : 본선 참가팀 중 가장 빠른 기록

4.  $T_{max}$ :  $T_{min}$ 의 1.45배.  $T_{your}$ 가  $T_{max}$ 를 초과할 경우  $T_{max}$ 로 간주.

③ 완주하지 못한 경우 주행한 거리를 기반으로 평가된다.

## 제6장 차년도 포물러 자율주행 경진대회 대비 규정

제21조 (차년도 포물러 자율주행 경진대회 대비 규정)

- ① 본 해 시뮬레이션 대회 참가자는 차년도 포물러 자율주행 대회(실차 기반) 참가 시 대회 보고서 점수에 가산점을 부여한다.
- ② 같은 팀의 판단 기준은 아래를 따른다.
  1. 같은 학교 및 같은 팀 명을 사용
  2. 참가인원의 과반 이상으로 구성