

OOD

The RVC Control SW

6팀 정상훈, 정수혁, 하재아, 한지훈

| 목차

Chapter 0. **수정된 Domain Model**

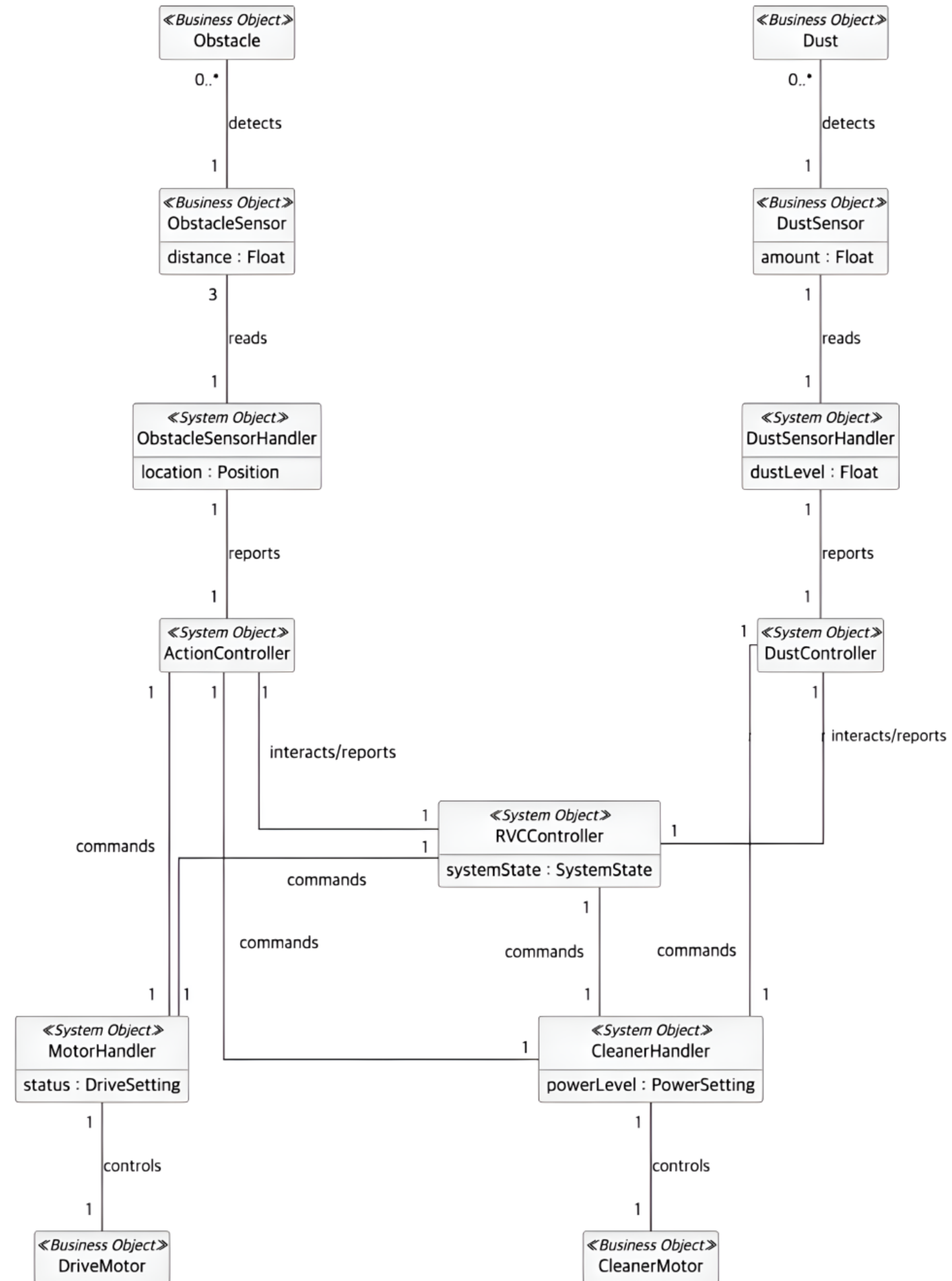
Chapter 1. **Use Cases (Fully dressed)**

Chapter 2. **Sequence Diagram**

Chapter 3. **Traceability Table**

Chapter 4. **Class Diagram**

수정된 Domain Model



UC-00 자동 청소 수행

Actor	User (Primary), Obstacle Sensor, Motor, Cleaner
Purpose	사용자가 청소를 시작하면 자동 청소를 수행하고, 전방 장애물이 감지되면 청소와 이동을 중지한다
Overview	User가 청소 시작을 요청하면 시스템은 Cleaner를 활성화하고 Motor를 전진시켜 자동 청소를 수행한다. 청소 도중 Obstacle Sensor가 전방 장애물을 감지하면 시스템은 Motor를 정지시키고 Cleaner를 비활성화한다. 이후 측면 회피(UC-01)한다
Cross Reference	R1.1, R1.2
Pre-Requisites	<ol style="list-style-type: none"> 1. 시스템 전원이 켜져 있어야 한다 2. Motor와 Cleaner가 정상 동작 가능한 상태여야 한다 3. Obstacle Sensor가 전방 장애물을 감지할 수 있는 상태여야 한다
Typical Courses of Events	<p>(A): Actor, (S): System</p> <ol style="list-style-type: none"> 1. (A) User가 자동 청소 시작을 위해 startCleaning()을 요청한다 2. (S) RVCController가 CleanerHandler의 activateCleaner()를 호출한다 3. (S) CleanerHandler가 Cleaner에 청소 시작(clean) 명령을 전달한다 4. (S) RVCController가 MotorHandler의 moveForward()를 호출한다 5. (S) MotorHandler가 Motor에 전진(move forward) 명령을 전달한다 6. (A) Obstacle Sensor가 장애물을 감지하면 감지한 위치 정보 sensorData(location)를 ObstacleSensorHandler에 전달한다 7. (S) ObstacleSensorHandler가 ActionController에 obstacleStatus(location)를 전달한다
Alternative Courses of Events	<p>A1. 전방 장애물이 감지된 경우</p> <ol style="list-style-type: none"> 1. (S) ActionController가 MotorHandler의 stopMotor()를 호출한다 2. (S) MotorHandler가 Motor에 정지(stop motor) 명령을 전달한다 3. (S) ActionController가 CleanerHandler의 deactivateCleaner()를 호출한다 4. (S) CleanerHandler가 Cleaner에 청소 중지(stop cleaning) 명령을 전달한다

UC-01 측면 회피

Actor	ObstacleSensor (Primary), Motor
Purpose	전방 장애물 감지 시 회피 로직을 수행한다
Overview	Obstacle Sensor가 넘겨준 sensorData의 값을 토대로ActionController가 회피할 방향을 결정해 Motor에게 지시하고 다시 청소를 시작한다. 좌,우 방향이 모두 막혀있으면 후진(uc-02)한다.
Cross Reference	R2.1, R2.2
Pre-Requisites	1. 전방 장애물이 감지된 상태이어야 한다 2. Motor 및 센서가 정상 동작 해야 한다
Typical Courses of Events	(A): Actor, (S): System 1. (A) Obstacle Sensor가 장애물을 감지하여 sensorData(location)을 전송한다 2. (S) ObstacleSensorHandler가 ActionController의 obstacleStatus(loation)을 호출한다 3. (S) ActionController가 MotorHandler의 turnLeft()를 호출한다 4. (S) MotorHandler가 Motor에 turn left 명령을 전달한다 5. (S) ActionController가 RVCController의 resumeClaening()을 호출한다
Alternative Courses of Events	A1. 좌측 장애물이 감지되고 우측에는 장애물이 없는 경우 1. (S) ActionController가 MotorHandler의 turnRight()를 호출한다 2. (S) MotorHandler가 Motor에 turn left 명령을 전달한다 3. (S) ActionController가 RVCController의 resumeClaening()을 호출한다 A2. 좌우측 모두 장애물이 감지된 경우 1. (S) ActionController가 RVCController의 moveBackward()을 호출한다

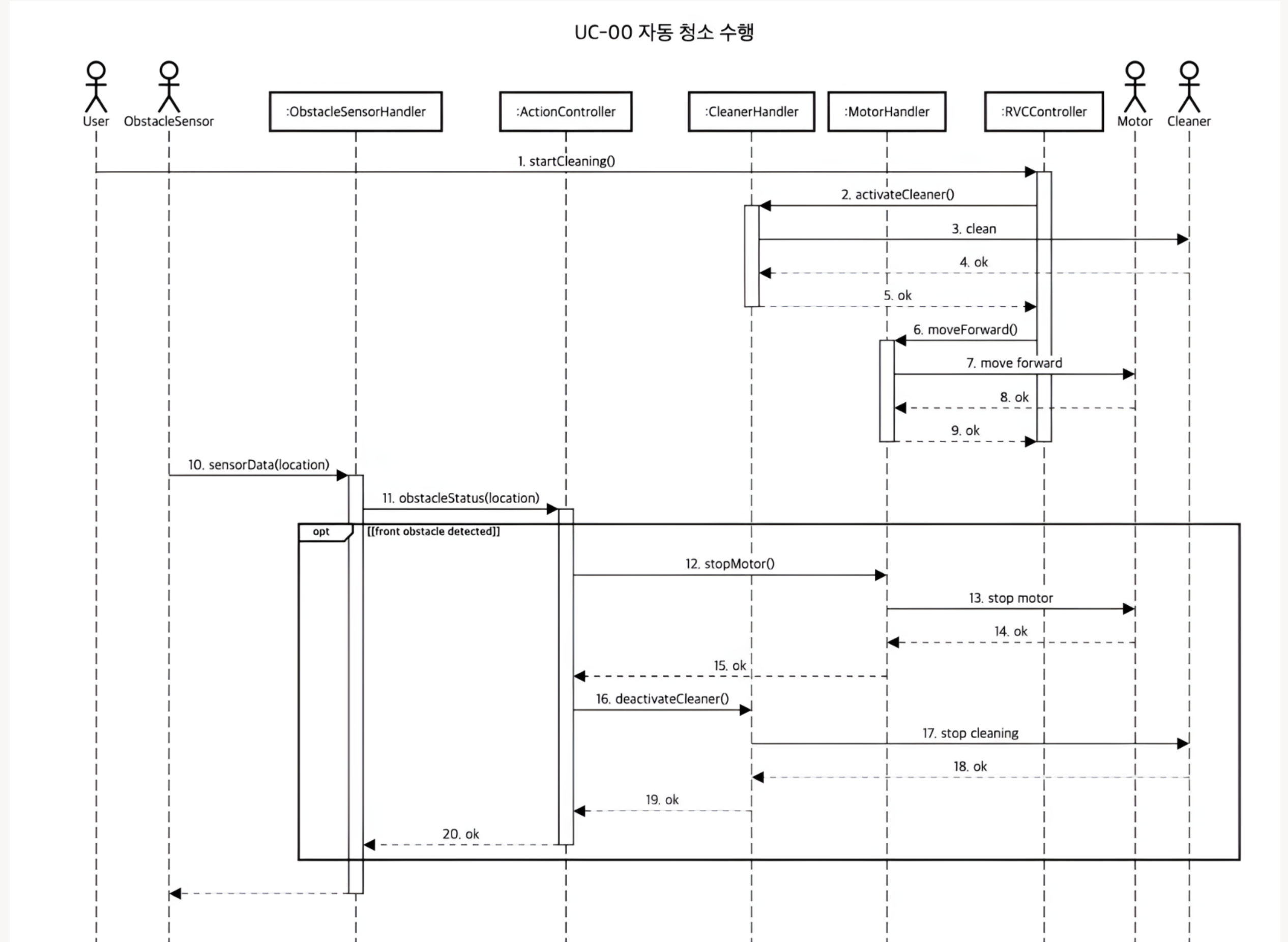
UC-02 후진 회피

Actor	ObstacleSensor (Primary), Motor
Purpose	전방 장애물로 인해 측면 회피가 불가능할 때 후진 회피를 수행한다
Overview	ObstacleSensor가 전방 장애물과 좌·우 회피 불가 상태를 감지하면 시스템은 Motor를 후진시킨다. 후진 중에는 좌·우 센서 상태를 계속 확인하며, 회피 가능한 방향이 확인되면 ActionController가 방향을 결정하여 Motor를 좌회전 또는 우회전시킨다. 이후 자동 청소로 복귀한다.
Cross Reference	R2.3, R2.4
Pre-Requisites	<ol style="list-style-type: none"> 1. 전방 장애물이 감지된 상태여야 한다 2. 좌/우 모두 장애물이 존재하거나 측면 회피가 불가능한 상태여야 한다 3. Motor 및 센서가 정상 동작해야 한다
Typical Courses of Events	<p>(A): Actor, (S): System</p> <ol style="list-style-type: none"> 1. (S) RVCController가 MotorHandler의 moveBackward()를 호출한다 2. (S) MotorHandler가 Motor에 move backward 명령을 전달한다 3. (A) ObstacleSensor가 후진 중 sensorData(location)를 전송한다 4. (S) ObstacleSensorHandler가 ActionController의 obstacleStatus(location)를 호출한다 5. (S) ActionController가 좌/우 상태를 판단한다. 6. (S) ActionController가 MotorHandler의 turnLeft()를 호출한다 7. (S) MotorHandler가 Motor에 turn left 명령을 전달한다 8. (S) ActionController가 RVCController의 resumeCleaning()을 호출한다
Alternative Courses of Events	<p>A1. 후진 중 우측이 회피 가능하다고 판단 된 경우</p> <ol style="list-style-type: none"> 1. (S) ActionController가 MotorHandler의 turnRight()를 호출한다 2. (S) MotorHandler가 Motor에 turn right 명령을 전달한다 3. (S) ActionController가 RVCController의 resumeCleaning()을 호출한다 <p>A2. 후진 중 좌/우 모두 계속 장애물로 판단 된 경우</p> <ol style="list-style-type: none"> 1. 시스템은 뚫린 방향이 감지될 때까지 후진과 상태 확인을 반복한다

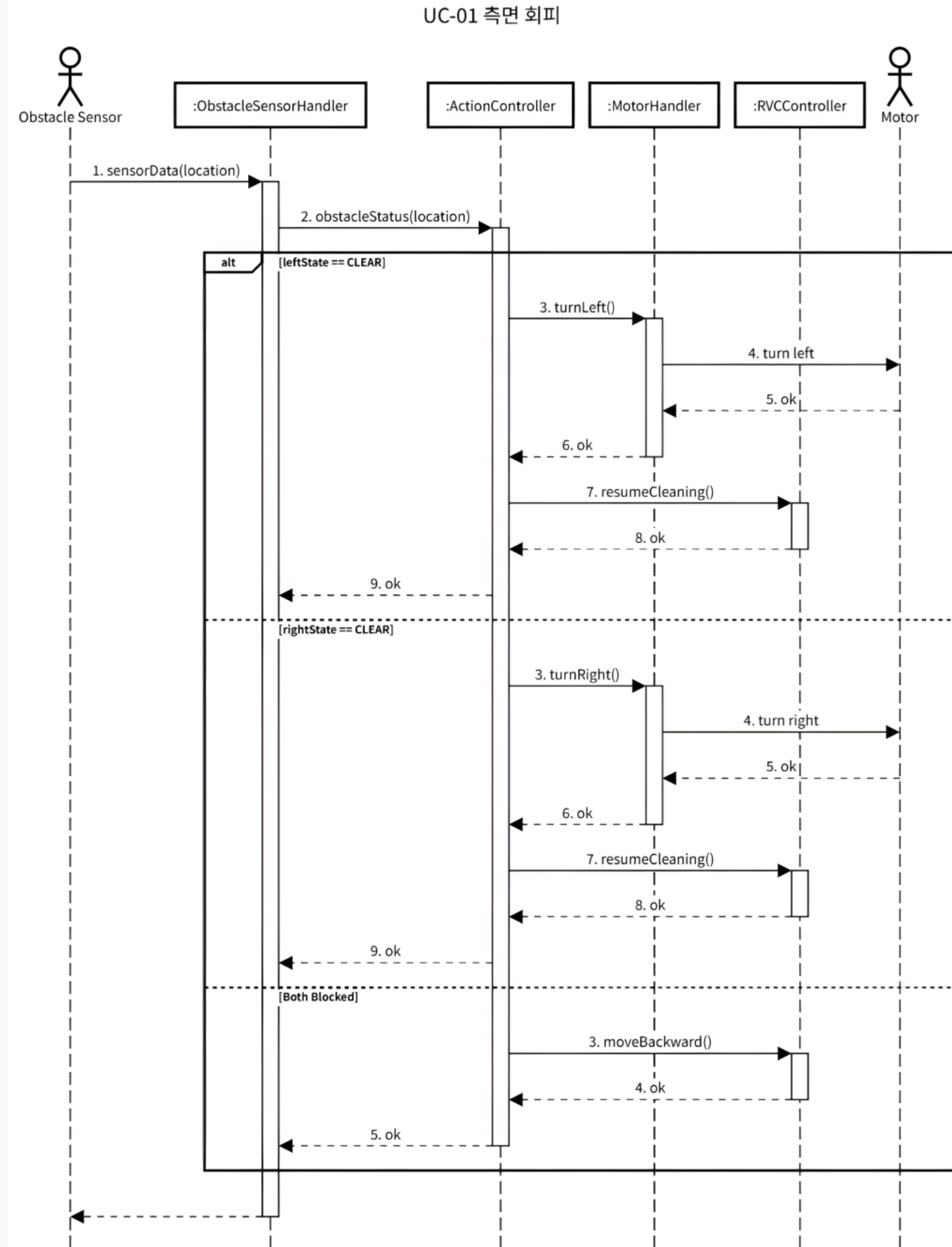
UC-03 집중 청소 수행

Actor	Dust Sensor (Primary), Cleaner
Purpose	먼지 감지 시 일정 시간 동안 청소 강도를 높여 집중 청소를 수행한다
Overview	Dust Sensor가 먼지를 감지하면 Cleaner를 강도 높은 모드(Up)로 전환하고 5초 타이머를 시작한다. 5초가 경과하면 일반 청소 모드(On)로 복귀한다
Cross Reference	R3.1, R3.2
Pre-Requisites	<ol style="list-style-type: none"> 1. 자동 청소(UC-00)가 진행 중이어야 한다 2. Dust Sensor가 먼지를 감지할 수 있는 상태여야 한다
Typical Courses of Events	<p>(A): Actor, (S): System</p> <ol style="list-style-type: none"> 1. (A) Dust Sensor가 먼지를 감지하여 sensorData(dustLevel)를 전송한다 2. (S) DustSensorHandler가 dustStatus(dustLevel)를 DustController에 전달한다 3. (S) DustController가 CleanerHandler에 boostPower()를 호출한다 4. (S) CleanerHandler가 Cleaner에 boost power 명령을 전달한다 5. (S) 5초 타이머가 경과한다 {delay = 5s} 6. (S) DustController가 CleanerHandler에 normalizePower()를 호출한다 7. (S) CleanerHandler가 Cleaner에 normalize power 명령을 전달한다 8. (S) DustController가 RVCController에 resumeNormalCleaning()을 호출한다
Alternative Courses of Events	없음

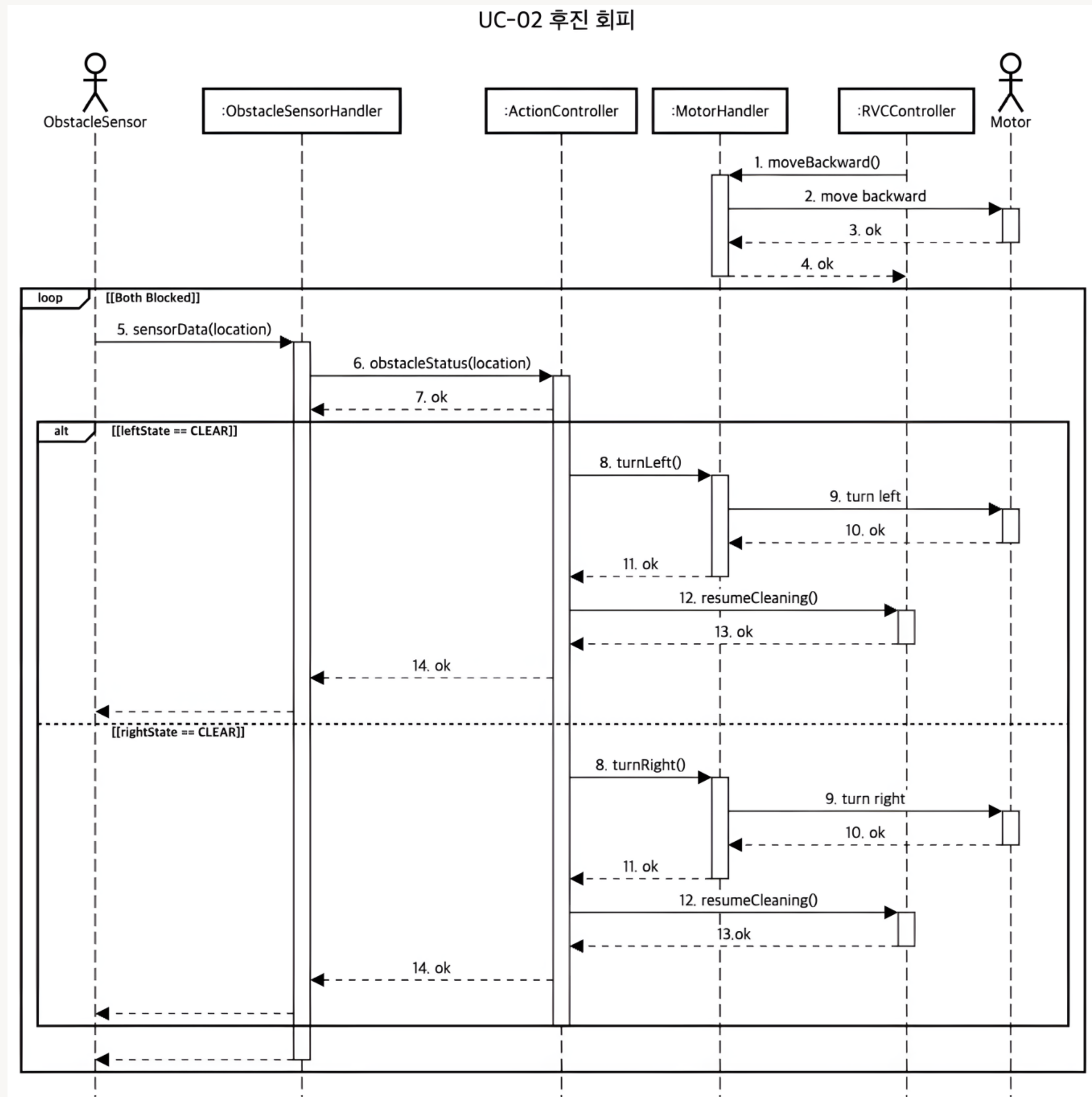
Sequence Diagram (UC-00)



Sequence Diagram (UC-01)

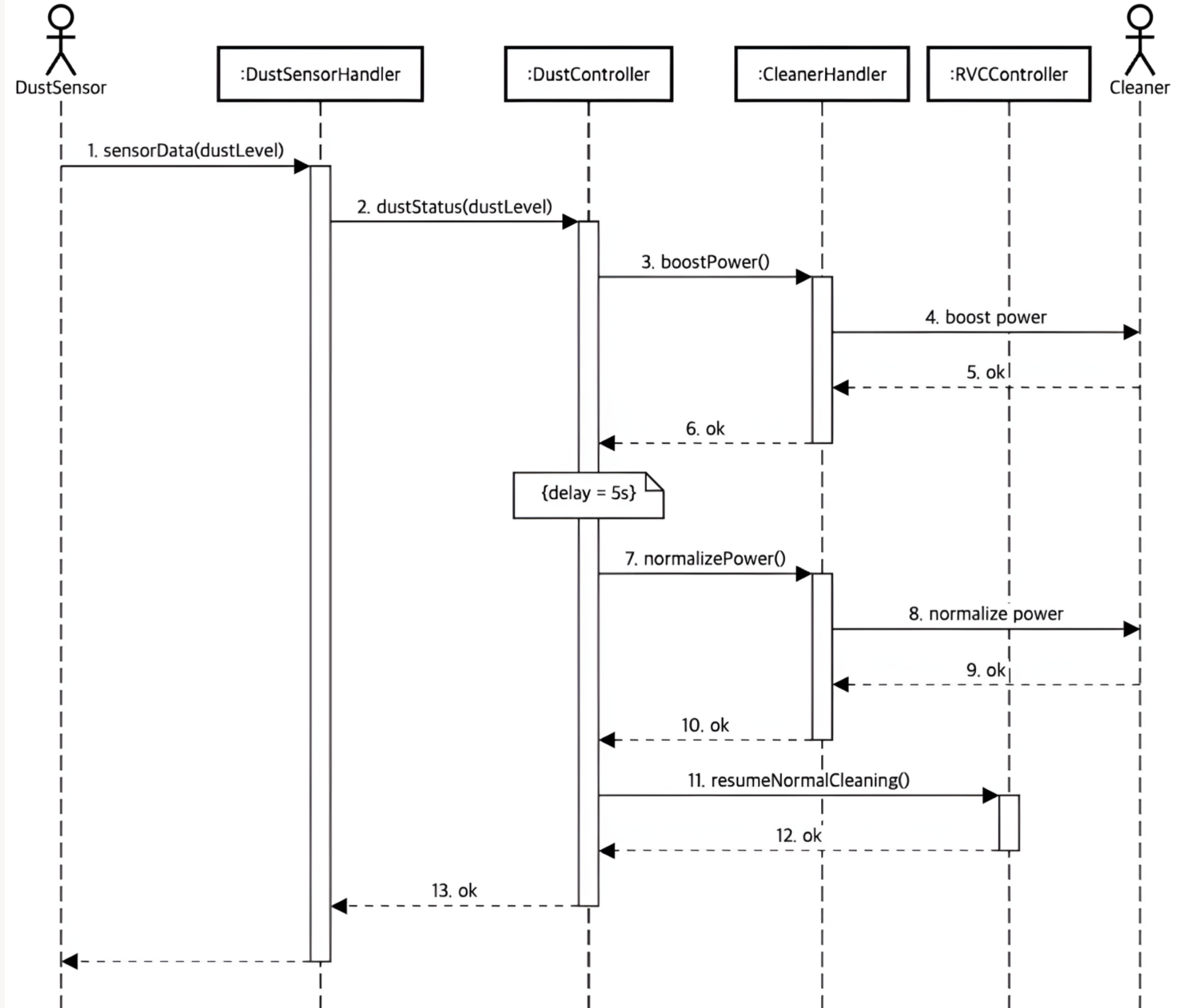


Sequence Diagram (UC-02)



Sequence Diagram (UC-03)

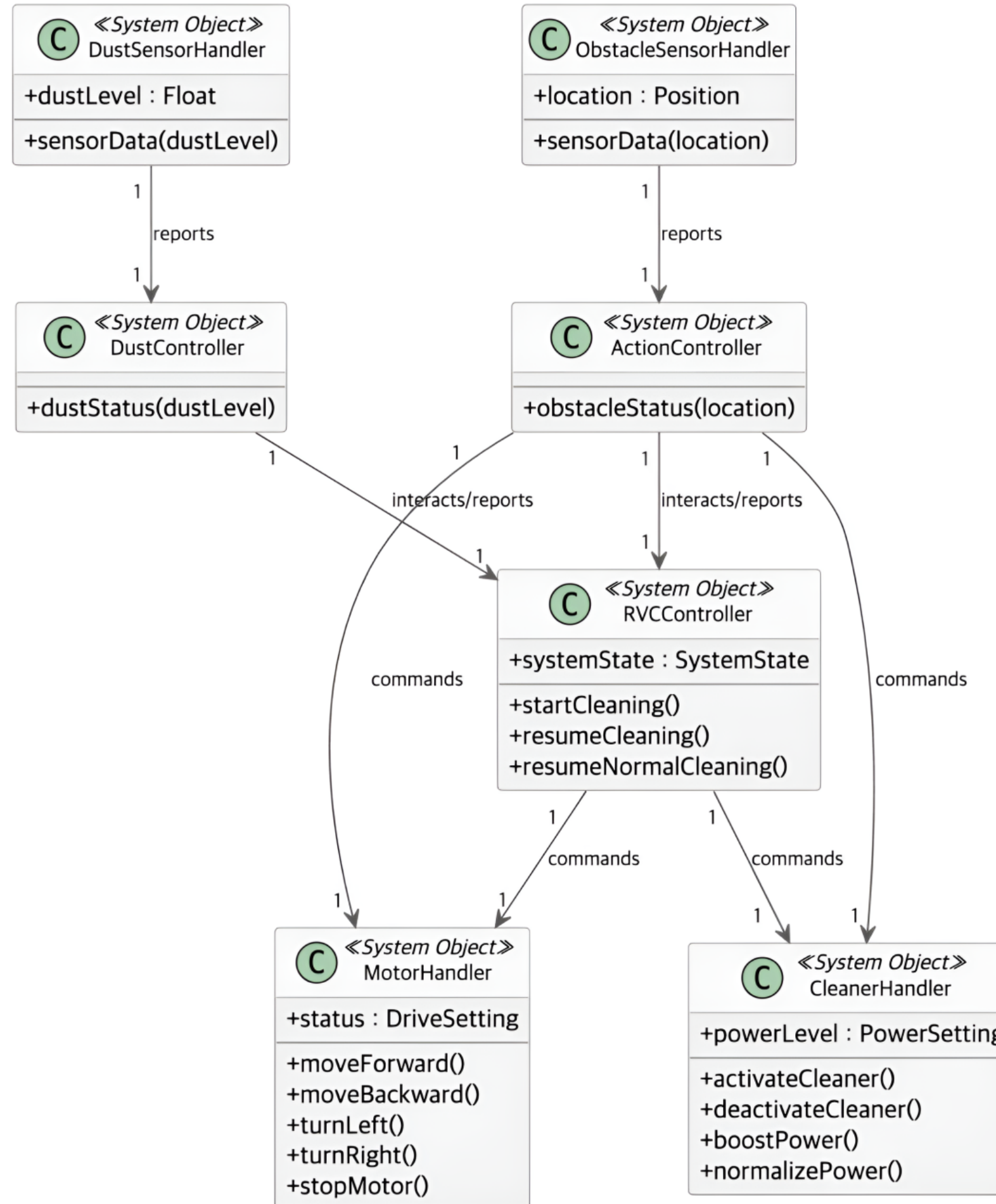
UC-03 집중 청소 수행



Traceability table

Use case		System Operation		Design Operation
UC-00	→	startCleaning()	→	startCleaning()
UC-01	→	notifyFrontObstacle()	→	resumeCleaning()
UC-02	→	notifySideObstacle()	→	resumeNormalCleaning()
UC-03	→	notifyDust()	→	sensorData(location)
			→	obstacleStatus(location)
			→	sensorData(dustLevel)
			→	dustStatus(dustLevel)
			→	moveForward()
			→	moveBackward()
			→	turnLeft()
			→	turnRight()
			→	stopMotor()
			→	activateCleaner()
			→	deactivateCleaner()
			→	boostPower()
			→	normalizePower()

Class Diagram



감사합니다

THANK YOU