

인터페이스 명세서

Roomie Confluence

19 August, 2025

- GUI 통신
 - RMS <-> AGUI
 - RMS <-> GGUI
 - RMS <-> SGUI
- ROS2 통신
 - RC <-> RMS
 - RC <-> RGUI
 - RC/AC <-> VS
 - RC <-> IOC
 - RC <-> AGUI
 - RC <-> AC
- serial 통신
 - AC <-> AD

From	To	통신 종류	통신 항목
GUI 통신			
Admin GUI	RMS	WebSocket, HTTP	<ul style="list-style-type: none"> • 동기 <ul style="list-style-type: none"> • 작업 목록 요청 <ul style="list-style-type: none"> • 시작 날짜 • 마지막 날짜 • 작업 유형 • 작업 상태 • 목적지 • 특정 작업 상세정보 요청 <ul style="list-style-type: none"> • 작업 ID • 로봇 목록 요청 <ul style="list-style-type: none"> • 로봇 ID • 모델명 • 로봇 상태 • 작업 상태

From	To	통신 종류	통신 항목
RMS	Admin GUI	WebSocket, HTTP	<ul style="list-style-type: none"> • 동기 <ul style="list-style-type: none"> • 작업 목록 응답 <ul style="list-style-type: none"> • 작업 ID • 작업 유형 • 작업 상태 • 목적지 • 로봇 ID • 작업 생성 시각 • 작업 종료 시각 • 특정 작업 상세정보 응답 <ul style="list-style-type: none"> • 로봇 할당 시각 • 적재 완료 시각 • 배송 도착 시각 • 작업 종료 시각 • 로봇 목록 응답 <ul style="list-style-type: none"> • 로봇 ID • 모델명 • 현재 위치 (구역) • 배터리 잔량 • 작업 상태 • 오류 여부 <ul style="list-style-type: none"> • Y • N • 오류 코드

From	To	통신 종류	통신 항목
			<ul style="list-style-type: none"> • 비동기 <ul style="list-style-type: none"> • 작업 상태 정보 업데이트 이벤트 <ul style="list-style-type: none"> • 총 작업 수 • 대기 작업 수 • 로봇 상태 정보 업데이트 이벤트 <ul style="list-style-type: none"> • 총 로봇 수 • 대기 로봇 수
Staff GUI	RMS	HTTP	<ul style="list-style-type: none"> • 동기 <ul style="list-style-type: none"> • 음식 주문 작업상태 전환 요청 <ul style="list-style-type: none"> • 작업 ID • 비품 주문 작업상태 전환 요청 <ul style="list-style-type: none"> • 작업 ID

From	To	통신 종류	통신 항목
RMS	Staff GUI	WebSocket, HTTP	<ul style="list-style-type: none"> • 동기 <ul style="list-style-type: none"> • 음식 주문 작업상태 전환 응답 <ul style="list-style-type: none"> • 작업 ID • 전환 상태 • 비품 주문 작업상태 전환 응답 <ul style="list-style-type: none"> • 작업 ID • 전환 상태 • 비동기 <ul style="list-style-type: none"> • 음식 주문 발생 이벤트 <ul style="list-style-type: none"> • 작업 ID • 주문내역 • 비품 주문 발생 이벤트 <ul style="list-style-type: none"> • 작업 ID • 요청 내역 • 음식 픽업 장소 도착 알림 이벤트 <ul style="list-style-type: none"> • 작업 ID • 로봇 ID • 비품 픽업 장소 도착 알림 이벤트 <ul style="list-style-type: none"> • 작업 ID • 로봇 ID

From	To	통신 종류	통신 항목
Guest GUI	RMS	HTTP	<ul style="list-style-type: none"> • 동기 <ul style="list-style-type: none"> • 호출 작업 생성 요청 <ul style="list-style-type: none"> • 요청 장소 • 작업 유형 • 음식 메뉴 요청 • 비품 목록 요청 • 배송 작업 생성 요청 <ul style="list-style-type: none"> • 요청 장소 • 작업 유형 • 주문 내역(음식/비품) • 호출 내역 조회 요청 <ul style="list-style-type: none"> • 요청 장소 • 호출 취소 요청 <ul style="list-style-type: none"> • 요청 장소 • 작업 ID • 주문 내역 조회 요청 <ul style="list-style-type: none"> • 요청 장소 • 작업 ID • 작업 유형

From	To	통신 종류	통신 항목
RMS	Guest GUI	WebSocket, HTTP	<ul style="list-style-type: none"> • 동기 <ul style="list-style-type: none"> • 호출 작업 생성 응답 <ul style="list-style-type: none"> • 요청 장소 • 작업 ID • 성공 여부 <ul style="list-style-type: none"> • Success • Fail • 작업 생성 시각 • 음식 메뉴 응답 <ul style="list-style-type: none"> • 매뉴명 • 가격 • 이미지 • 비품 목록 응답 <ul style="list-style-type: none"> • 비품명 • 이미지 • 배송 작업 생성 응답 <ul style="list-style-type: none"> • 요청 장소 • 작업 ID • 성공 여부 <ul style="list-style-type: none"> • Success • Fail • 작업 생성 시각 • 호출 내역 조회 응답 <ul style="list-style-type: none"> • 요청 장소

From	To	통신 종류	통신 항목
			<ul style="list-style-type: none"> • 작업 ID • 예상 소요 시간 • 호출 취소 응답 <ul style="list-style-type: none"> • 요청 장소 • 작업 ID • 성공 여부 <ul style="list-style-type: none"> • Success • Fail • 취소 처리 시각 • 주문 내역 조회 응답 <ul style="list-style-type: none"> • 요청 장소 • 작업 ID • 작업 유형 • 예상 소요 시간 • 작업 상세내역 • 비동기 <ul style="list-style-type: none"> • 호출 가능 로봇 확인중 이벤트 • 호출 요청 접수 완료 이벤트 <ul style="list-style-type: none"> • 작업 ID • 예상 대기 시간 • 로봇 도착 완료 알림 이벤트 <ul style="list-style-type: none"> • 작업 ID • 요청 장소

From	To	통신 종류	통신 항목
			<ul style="list-style-type: none"> • 배송 완료 알림 이벤트 <ul style="list-style-type: none"> • 작업 ID • 요청 장소 • 시간초과 복귀 알림 이벤트 <ul style="list-style-type: none"> • 작업 ID • 요청 장소
RC	Admin GUI	ROS2	<ul style="list-style-type: none"> • 비동기 <ul style="list-style-type: none"> • 로봇 ID • 배터리 잔량 • 현재 위치 • 현재 방향

From	To	통신 종류	통신 항목
Roomie 외부 통신			
RC	RMS	ROS2	<ul style="list-style-type: none"> • 비동기 <ul style="list-style-type: none"> • 로봇 상태 동기화 <ul style="list-style-type: none"> • 로봇 ID • 현재 상태 • 요청 상태 • 도착 <ul style="list-style-type: none"> • 로봇 ID • 작업 ID • 도착 장소 <ul style="list-style-type: none"> • 픽업장소 • 배송장소 • 대기장소 • 카운트다운 종료 • 배터리 잔량 • 현재 위치 • 현재 방향 • 작업 생성 요청 <ul style="list-style-type: none"> • 목적지

From	To	통신 종류	통신 항목
RMS	RC	ROS2	<ul style="list-style-type: none"> 비동기 <ul style="list-style-type: none"> 작업 할당 지시 <ul style="list-style-type: none"> 작업 ID 작업 유형 주문정보 복귀 지시 작업 상태 동기화 지시
Roomie 내부 통신 (MVP)			
RC	AC	Serial	<ul style="list-style-type: none">
AC	RC	Serial	<ul style="list-style-type: none"> 버튼 클릭 요청 <ul style="list-style-type: none"> 버튼 종류 좌표

From	To	통신 종류	통신 항목
RC	IC	Serial	<ul style="list-style-type: none"> • 비동기 <ul style="list-style-type: none"> • 완료 <ul style="list-style-type: none"> • 카드키 인식 <ul style="list-style-type: none"> • 호실 번호 • 성공 여부 <ul style="list-style-type: none"> • Success • Fail • 도어 잠금 <ul style="list-style-type: none"> • 성공 여부 <ul style="list-style-type: none"> • Success • Fail • 도어 잠금 해제 <ul style="list-style-type: none"> • 성공 여부 <ul style="list-style-type: none"> • Success • Fail • 도어 닫힘 <ul style="list-style-type: none"> • 여부 <ul style="list-style-type: none"> • True (닫힘) • False (열림) • 화물 적재 <ul style="list-style-type: none"> • 여부 <ul style="list-style-type: none"> • True (적재됨) • False (적재되지 않음)

From	To	통신 종류	통신 항목
			<ul style="list-style-type: none"> • 실시간 <ul style="list-style-type: none"> • 로봇 자세
IC	RC	Serial	<ul style="list-style-type: none"> • 동기 <ul style="list-style-type: none"> • 카드키 인식 요청 • 도어 잠금 해제 요청 • 하차 여부 감지 요청 • 도어 닫힘 감지 요청

From	To	통신 종류	통신 항목
RC	VS	ROS2	<ul style="list-style-type: none"> • 동기 <ul style="list-style-type: none"> • • 비동기 <ul style="list-style-type: none"> • 인식 모드 전환 완료 <ul style="list-style-type: none"> • 인식 모드 ID <ul style="list-style-type: none"> • 상하 감지 모드 • 속도 변화 요청 • 감지 결과 <ul style="list-style-type: none"> • 버튼 감지 정보 전송 <ul style="list-style-type: none"> • 버튼이름 • 좌표 • ON,OFF상태 • 엘리베이터 입구 너비 감지결과 <ul style="list-style-type: none"> • 좌측 경계 좌표 • 우측 경계 좌표 • 엘리베이터 위치 및 방향 감지 결과 • 문 열림 감지 결과 • 탑승/하차 확보 공간 감지 결과 • 장소 감지 결과 <ul style="list-style-type: none"> • 장소 ID

From	To	통신 종류	통신 항목
VS	RC	ROS2	<ul style="list-style-type: none"> • 동기 <ul style="list-style-type: none"> • 인식 모드 전환 요청 <ul style="list-style-type: none"> • 인식 모드 ID • 엘리베이터 <ul style="list-style-type: none"> • 버튼 감지 요청 • 문 열림 상태 요청 • 내부 공간 요청 • 장소감지 요청 • 비동기 <ul style="list-style-type: none"> • 길안내 중 이벤트 <ul style="list-style-type: none"> • 감속 (멀어짐) • 정속 (거리 유지) • 이탈 (사라짐) • 주행 재개 (복귀)
RC	Robot GUI	ROS2	<ul style="list-style-type: none"> • 동기 <ul style="list-style-type: none"> • 도어 잠금 해제 요청 • 비동기 <ul style="list-style-type: none"> • 카드키 인식 시작 • 목적지 직접 입력 <ul style="list-style-type: none"> • 목적지 • 출발 카운트다운 종료 • 수령 완료 클릭

From	To	통신 종류	통신 항목
Robot GUI	RC	ROS2	<ul style="list-style-type: none"> • 동기 <ul style="list-style-type: none"> • GUI 업데이트 <ul style="list-style-type: none"> • 출발 카운트다운 시작 화면 • 복귀 카운트다운 시작 화면 • 비동기 <ul style="list-style-type: none"> • GUI 업데이트 <ul style="list-style-type: none"> • 호출이동중 화면 • 길 안내 <ul style="list-style-type: none"> • 사람 인식 화면 • 길안내 완료 화면 • 주문 <ul style="list-style-type: none"> • 픽업 장소 이동중 화면 • 주문요청 상세 화면 • 배송도착 초기 화면 • 기본 • '수령완료' 버튼 표시 • 엘리베이터 <ul style="list-style-type: none"> • 버튼 조작중 화면 • 호출중 화면 • 탑승중 화면 • 목적층 이동중 화면 • 하차중 화면 • 충전중 화면

From	To	통신 종류	통신 항목
			<ul style="list-style-type: none"> 로봇 상태 변화 요청 <ul style="list-style-type: none"> 사용자 점유 인식 모드 전환 요청 <ul style="list-style-type: none"> 인식 모드 ID 안내 주행 시작 안내 주행 재시작
VS	Robot GUI	UDP	<ul style="list-style-type: none"> 비동기 <ul style="list-style-type: none"> 영상

1 GUI 통신

- [RMS <-> AGUI](#)
- [RMS <-> GGUI](#)
- [RMS <-> SGUI](#)

1.1 RMS <-> AGUI

HTTP					
API 명	HTTP 메서드	URL	요청 데이터 구조	응답 데이터 구조	비고

HTTP					
작업 목록 조회	POST	/api/gui/ task_list	필터 미적용 시 <pre>{ "type": "request", "action": "task_list", "payload": { "filters": {} // 빈 필터 = 전체 조회 } }</pre> 필터 적용 시 <pre>{ "type": "request", "action": "task_list", "payload": { "filters": { "start_date": "2024-01-01", //str "end_date": "2024-01-02", //str "task_type": "음식배송", //str "task_status": "수령 완료", //str "destination": "ROOM_102" //str } } }</pre>	<pre>{ "type": "response", "action": "task_list", "payload": { "tasks": [{ "task_id": 12, //int "task_type": "음식배송", //str "task_status": "수령 완료", //str "destination": "ROOM_202", //str "robot_id": 1, //int "task_creation_time": "2024-03-15T10:30:00Z", //ISO 8601, KST "task_completion_time": "2024-03-15T10:52:10Z" //ISO 8601, KST }, { "task_id": "13", "task_type": "음식배송", "task_status": "픽업 대기 중", "destination": "ROOM_102", "robot_id": 2, "task_creation_time": "2024-03-15T10:30:00Z", "task_completion_time": null }] } }</pre>	task_type <ul style="list-style-type: none"> 0: 음식배송 1: 비품배송 2: 호출 3: 길안내 task_status <ul style="list-style-type: none"> 0: 접수됨 1: 준비 완료 2: 로봇 할당됨 3: 픽업 장소로 이동 4: 픽업 대기 중 5: 배송 중 6: 배송 도착 7: 수령 완료 10: 호출 이동 중 11: 호출 도착 20: 길안내 접수됨 21: 길안내 중 22: 길안내 도착

HTTP					
API 명	HTTP 메서드	URL	요청 데이터 구조	응답 데이터 구조	비고
특정 작업 상세정보 조회	POST	/api/gui/task_detail	<pre>{ "type": "request", "action": "task_detail", "payload": { "task_id": 12 //int } }</pre>	<pre>{ "type": "response", "action": "task_detail", "payload": { "robot_assignment_time": "2024-03-15T10:32:00Z", //ISO 8601, KST "pickup_completion_time": "2024-03-15T10:45:00Z", //ISO 8601, KST "delivery_arrival_time": null, //ISO 8601, KST "task_completion_time": null //ISO 8601, KST } }</pre>	

HTTP					
API 명	HTTP 메서드	URL	요청 데이터 구조	응답 데이터 구조	비고

HTTP					
로봇 목록 조회	POST	<code>/api/gui/ robot_list</code>	<p>필터 미적용 시</p> <pre>{ "type": "request", "action": "robot_list", "payload": { "filters": {} // 빈 필터 = 전체 조회 } }</pre> <p>필터 적용 시</p> <pre>{ "type": "request", "action": "robot_list", "payload": { "filters": { "robot_id": 1, //str "model_name": "ServiceBot_V2", //str "robot_status": "복귀 중" //str } } }</pre>	<pre>{ "type": "response", "action": "robot_list", "payload": { "robots": [{ "robot_id": 1, "model_name": "ServiceBot_V2", "battery_level": 85, "is_charging": false, "robot_status": "배송장소 이동", "task_id": 13, "has_error": false, "error_code": null }, { "robot_id": 2, "model_name": "ServiceBot_V1", "battery_level": 45, "is_charging": true, "robot_status": "충전상태", "task_id": null, "has_error": false, "error_code": null }] } }</pre>	<p>location</p> <ul style="list-style-type: none"> • 0: LOB_WAITING • 1: LOB_CALL • 2: RES_PICKUP • 3: RES_CALL • 4: SUP_PICKUP • 5: ELE_1 • 6: ELE_2 • 101: ROOM_101 • 102: ROOM_102 • 201: ROOM_201 • 202: ROOM_202

WebSocket				
이벤트명	방향	URL	전송 데이터 구조	비고
작업 상태 정보 업데이트	RMS → AGUI	/api/gui/ws/admin/ {admin_id}	{ "type": "event", "action": "task_status_update", "payload": { "total_task_count": 25, //int "waiting_task_count": 3 //int } }	

WebSocket				
이벤트명	방향	URL	전송 데이터 구조	비고
로봇 상태 정보 업데이트	RMS → AGUI	/api/gui/ws/admin/ {admin_id}	{ "type": "event", "action": "robot_status_update", "payload": { "total_robot_count": 5, //int "active_robot_count": 2 //int } }	

1.2 RMS <-> GGUI

HTTP					
API 명	HTTP 메서드	URL	요청 데이터 구조	응답 데이터 구조	비고
호출 작업 생성	POST	/api/gui/ create_call_task	<pre>{ "type": "request", "action": "create_call_task", "payload": { "location_name": "ROOM_201", // str "task_type_id": 2 // int, 0~3, 필수 (0: 음식, 1: 비품, 2: 호출, 3: 길안내) } }</pre>	<pre>{ "type": "response", "action": "create_call_task", "payload": { "location_name": "ROOM_201", //str "task_name": "TASK_006", //str "success": true, //bool "error_code": null, // int, 실패 시 오류 코드 "error_message": null // str, 실패 시 오류 메시지 }, "task_creation_time": 2025-07-22T16:42:16+09:00 //DATETIME ISO 8601,kst }</pre>	location <ul style="list-style-type: none"> • 0: LOB_WAITING • 1: LOB_CALL • 2: RES_PICKUP • 3: RES_CALL • 4: SUP_PICKUP • 5: ELE_1 • 6: ELE_2 • 101: ROOM_101 • 102: ROOM_102 • 201: ROOM_201 • 202: ROOM_202 task_type <ul style="list-style-type: none"> • 0: 음식배송 • 1: 비품배송 • 2: 호출 • 3: 길안내

HTTP					
API 명	HTTP 메서드	URL	요청 데이터 구조	응답 데이터 구조	비고
호출 내역 조회	POST	/api/gui/get_call_history	<pre>{ "type": "request", "action": "get_call_history", "payload": { "location_name": "ROOM_102" //str "task_name": "TASK_006" //str } }</pre>	<pre>{ "type": "response", "action": "get_call_history", "payload": { "location_name": "ROOM_102", "task_name": "TASK_006", //str "task_type_name": "호출", //str "estimated_time": 5 //int "robot_status" : { "x": 0.2, //float "y": 1.2, //float "floor_id": 1 //int } } }</pre>	

HTTP					
API 명	HTTP 메서드	URL	요청 데이터 구조	응답 데이터 구조	비고

HTTP					
음식 메뉴 조회	POST	/api/gui/ get_food_menu	{ "type": "request", "action": "get_food_menu", "payload": { "location_name": "ROOM_201" //str } }	{ "type": "response", "action": "get_food_menu", "payload": { "food_items": [//array { "food_name": "스파게티", //str "price": 15000, //int "image": "data:image/jpeg;base64,/9j/4AAQSk.. (생략)" // }, { "food_name": "피자", "price": 25000, "image": "data:image/jpeg;base64,/9j/4AAQSk.. (생략)" } } }	Food <ul style="list-style-type: none"> • 0: 스파게티 • 1: 피자 • 2: 스테이크 • 3: 버거 Supply <ul style="list-style-type: none"> • 0: 칫솔 • 1: 타월 • 2: 생수 • 3: 수저

HTTP					
API 명	HTTP 메서드	URL	요청 데이터 구조	응답 데이터 구조	비고

HTTP					
배송 작업 생성	POST	/api/gui/ create_delivery_task	{ "type": "request", "action": "create_delivery_task", "payload": { "location_name": "ROOM_201", // "task_type_name": "음식배달", "order_details": { "items": [//array { "name": "스파게티", //str "quantity": 2, //int "price": 15000 //int }, { "name": "피자", "quantity": 1, "price": 15000 } } } }	{ "type": "response", "action": "create_delivery_task", "payload": { "location_name": "ROOM_201", //str "task_name": "TASK_005", //str "success": true, //bool "error_code": null, // int, 실패 시 오류 코드 "error_message": null // str, 실패 시 오류 메시지 "estimated_time": 55, //int 도착 예상 시간 "task_creation_time": 2025-07-22T16:42:16+09:00 //DATETIME ISO 8601, kst } }	

HTTP					
API 명	HTTP 메서드	URL	요청 데이터 구조	응답 데이터 구조	비고

HTTP					
요청 내역 조회	POST	/api/gui/ get_task_list	{ "type": "request", "action": "get_task_list", "payload": { "location_name": "ROOM_201" } }	{ "type": "response", "action": "get_task_list", "payload": { "location_name": "ROOM_201", // string "tasks": [{ "task_name": "TASK_001", "task_type_name": "음식배송", "created_at": "2025-07- 22T16:42:16+09:00" }, { "task_name": "TASK_002", "task_type_name": "로봇호출", "created_at": "2025-07-22T17:03:15+09:00" }] } ... }	

HTTP					
API 명	HTTP 메서드	URL	요청 데이터 구조	응답 데이터 구조	비고

HTTP				
주문 상세내역 조회	POST	<code>/api/gui/ get_order_his tory</code>	<pre>{ "type": "request", "action": "get_order_history", "payload": { "request_location": "ROOM_201", "task_name": "TASK_001", "task_type_name": "음식배달" } }</pre>	<pre>{ "type": "response", "action": "get_order_history", "payload": { "request_location": "ROOM_201", //str "task_name": "TASK_001", //str "task_type_name": "음식배달", //str "estimated_time": 55, //int "task_creation_time": "2025-07-22T16:42:16+09:00", //DATETIME ISO 8601,kst "robot_assignment_time": "2025-07-22T16:42:16+09:00", //DATETIME ISO 8601,kst "pickup_completion_time": "2025-07-22T16:42:16+09:00", //DATETIME ISO 8601,kst "delivery_arrival_time": null, //DATETIME ISO 8601,kst } }</pre>
WebSocket				
이벤트명	방향	URL	전송 데이터 구조	비고

WebSocket				
호출 수락 알림	RMS → GGUI	<code>/api/gui/ws/guest/ {location_name}</code>	<pre>{ "type": "event", "action": "call_request_acceptance", "payload": { "task_name": "TASK_006", //str "estimated_wait_time": 15 //int } }</pre>	

WebSocket				
이벤트명	방향	URL	전송 데이터 구조	비고
로봇 도착 완료 알림	RMS → GGUI	<code>/api/gui/ws/guest/ {location_name}</code>	<pre>{ "type": "event", "action": "robot_arrival_completion", "payload": { "task_name": "TASK_006", //str "location_name": "ROOM_102" //str } }</pre>	

WebSocket				
이벤트명	방향	URL	전송 데이터 구조	비고

WebSocket				
배송 완료 알림	RMS → GGUI	<code>/api/gui/ws/guest/ {location_name}</code>	<pre>{ "type": "event", "action": "delivery_completion", "payload": { "task_name": "TASK_001", //str "request_location": "ROOM_102" //str } }</pre>	

WebSocket				
이벤트명	방향	URL	전송 데이터 구조	비고
로봇 시간초과 복귀 알림	RMS → GGUI	<code>/api/gui/ws/guest/ {location_name}</code>	<pre>{ "type": "event", "action": "task_timeout_return", "payload": { "task_name": "TASK_006", //str "location_name": "ROOM_102" //str } }</pre>	

| |

1.3 RMS <-> SGUI

HTTP					
API 명	HTTP 메서드	URL	요청 데이터 구조	응답 데이터 구조	비고
음식 주문 작업상태 전환	POST	api/gui/ food_order_status_change	{ "type": "request", "action": "food_order_status_change", "payload": { "task_id": 12 //int } }	{ "type": "response", "action": "food_order_status_change", "payload": { "task_id": 12, //int "status_changed": "food_ready" //str } }	

WebSocket				
이벤트명	방향	URL	전송 데이터 구조	비고

WebSocket				
음식 주문 발생 알림	RMS → SGUI	<code>/api/gui/ws/staff/ {staff_id}</code>	<pre>{ "type": "event", "action": "food_order_creation", "payload": { "task_id": 12, //int "request_location": "ROOM_307", //str "order_details": { "items": [{ "name": "스파게티", "quantity": 2, //int "price": 15000 //int }, { "name": "피자", //str "quantity": 1, //int "price": 15000 //int }] } } }</pre>	food <ul style="list-style-type: none"> • 0: 스파게티 • 1: 피자 • 2: 스테이크 • 3: 버거

WebSocket				
이벤트명	방향	URL	전송 데이터 구조	비고

WebSocket				
음식 픽업장소 도착 알림	RMS → SGUI	<code>/api/gui/ws/staff/ {staff_id}</code>	<code>{ "type": "event", "action": "food_pickup_arrival", "payload": { "task_id": 12, //int "robot_id": 1 //int } }</code>	

WebSocket				
이벤트명	방향	URL	전송 데이터 구조	비고
음식 배송장소 도착 알림	RMS → SGUI	<code>/api/gui/ws/staff/ {staff_id}</code>	<code>{ "type": "event", "action": "food_delivery_arrival", "payload": { "task_id": 12, //int "robot_id": 1 //int } }</code>	

WebSocket				
이벤트명	방향	URL	전송 데이터 구조	비고

WebSocket				
비품 요청 발생 알림	RMS → SGUI	<code>/api/gui/ws/staff/ {staff_id}</code>	<pre>{ "type": "event", "action": "supply_order_creation", "payload": { "task_id": 12, //int "request_location": "ROOM_307", //str "request_details": { "items": [{ "name": "타월", //str "quantity": 3 //int }, { "name": "생수", //str "quantity": 2 //int }] } } }</pre>	supply <ul style="list-style-type: none"> • 0: 칫솔 • 1: 타월 • 2: 생수 • 3: 수저
WebSocket				
이벤트명	방향	URL	전송 데이터 구조	비고

WebSocket				
비품 픽업장소 도착 알림	RMS → SGUI	<pre> /api/gui/ws/staff/ {staff_id} </pre>	<pre> { "type": "event", "action": "supply_pickup_arrival", "payload": { "task_id": 12, //int "robot_id": 1 //int } } </pre>	


○○○

2 ROS2 통신

- RC <-> RMS
- RC <-> RGUI
- RC/AC <-> VS
- RC <-> IOC
- RC <-> AGUI
- RC <-> AC

2.1 RC <-> RMS

From	To	프로토콜	인터페이스 항목	메시지 형식	
Action					

From	To	프로토콜	인터페이스 항목	메시지 형식	
RMS	RC	Action	작업 할당 <code>/roomie/action/ perform_task</code>	<pre> # PerformTask.action # Goal int32 robot_id int32 task_id int32 task_type_id int32 target_location_id int32 pickup_location_id string order_info # 주문 정보 (JSON) --- # Result (RCS -> RMS 최종 결과) int32 robot_id int32 task_id bool success string message --- # Feedback (RCS(수행중) -> RMS 진행 상황) int32 robot_id int32 task_id </pre>	<p>task_type_id</p> <ul style="list-style-type: none"> • 0: 음식배송 • 1: 비품배송 • 2: 호출 • 3: 길안내 <p>location_id</p> <ul style="list-style-type: none"> • 0: LOB_WAITING • 1: LOB_CALL • 2: RES_PICKUP • 3: RES_CALL • 4: SUP_PICKUP • 5: ELE_1 • 6: ELE_2 • 101: ROOM_101 • 102: ROOM_102 • 201: ROOM_201 • 202: ROOM_202 <p>order_info</p> <div>  room_number 추가했습니다. </div>

From	To	프로토콜	인터페이스 항목	메시지 형식	
					<pre> # 주문 정보 "{ "room_number": "202", "items": [{ "name": "스파게티", "quantity": 2 }, { "name": "피자", "quantity": 1 }] }" </pre>

From	To	프로토콜	인터페이스 항목	메시지 형식	
RMS	RC	Action	복귀 <code>/roomie/action/ perform_return</code>	<pre> # PerformReturn.action # Goal int32 robot_id --- # Result int32 robot_id string message --- # Feedback (없음) </pre>	
Service					

From	To	프로토콜	인터페이스 항목	메시지 형식	
RC	RMS	Service	좌표 데이터 요청 /roomie/command/ get_locations	<pre> # GetLocations.srv # Request int32 robot_id --- # Response int32 robot_id bool success int32[] location_ids int32[] floor_ids float32[] location_xs float32[] location_ys </pre>	
Topic					

From	To	프로토콜	인터페이스 항목	메시지 형식	
RC	RMS	Topic	로봇 상태 /roomie/status/ robot_state	<pre># RobotState.msg int32 robot_id int32 robot_state_id</pre>	robot_state_id <ul style="list-style-type: none"> • 0: 초기화 • 1: 충전상태 • 2: 작업대기 • 10: 픽업위치 이동 • 11: 픽업대기 • 12: 배송장소 이동 • 13: 수령대기 • 20: 호출위치 이동 • 21: 길안내 목적지 입력대기 • 22: 길안내 이동 • 23: 대상 탐색 • 30: 대기위치로 이동 • 31: 엘리베이터 탑승 • 90: 오류
RC	RMS	Topic	배터리 잔량 /roomie/status/ battery_status	<pre># BatteryStatus.msg int32 robot_id float32 charge_percentage bool is_charging</pre>	

From	To	프로토콜	인터페이스 항목	메시지 형식	
RC	RMS	Topic	현재 로봇 위치와 방향 /roomie/status/ roomie_pose	<pre># RoomiePose.msg int32 robot_id int32 floor_id geometry_msgs/Pose pose</pre>	floor_id <ul style="list-style-type: none"> • 0: 1층 • 1: 2층 • 2: 3층 • 3: 4층

2.2 RC <-> RGUI

From	To	인터페이스 항목	메시지 형식	비고
ROS2 Action				
RC	RGUI	출발 카운트다운 <code>/robot_gui/action/ start_countdown</code>	<pre> # StartCountdown.action # Goal int32 robot_id int32 task_id int32 task_type_id --- # Result int32 robot_id bool success --- # Feedback int32 remaining_time # 초 단위 남은 시간 </pre>	task_type_id <ul style="list-style-type: none"> • 0: 음식배송 • 1: 비품배송 • 2: 호출 • 3: 길안내

From	To	인터페이스 항목	메시지 형식	비고
RC	RGUI	복귀 카운트다운 /robot_gui/action/ return_countdown	<pre> # ReturnCountdown.action # Goal int32 robot_id --- # Result int32 robot_id bool success --- # Feedback int32 remaining_time # 초 단위 남은 시간 </pre>	
ROS2 Topic				

From	To	인터페이스 항목	메시지 형식	비고
RC	Robot GUI	로봇 GUI 이벤트 전송 <code>/robot_gui/event</code>	<pre># RobotGuiEvent.msg int32 robot_id int32 rgui_event_id int32 task_id builtin_interfaces/Time timestamp string detail</pre>	<p>rgui_event_id</p> <ul style="list-style-type: none"> 1: 엘리베이터 버튼 조작 시작 2: 엘리베이터 버튼 조작 종료 3: 엘리베이터 탑승 시작 4: 엘리베이터 탑승 종료 5: 엘리베이터 하차 시작 6: 엘리베이터 하차 종료 7: 호출 이동 시작 8: 호출 이동 종료 9: 호실 번호 인식 완료 <ul style="list-style-type: none"> detail <div># 호실 번호 "101"</div> 10: 길안내 이동 시작 11: 길안내 이동 종료 12: 픽업장소 이동 시작 13: 픽업장소 이동 종료 <ul style="list-style-type: none"> detail <div># 주문 정보</div>

From	To	인터페이스 항목	메시지 형식	비고
				<pre> "{ "room_number": "202", "items": [{ "name": "스파게 티", "quantity": 2 }, { "name": "피자", "quantity": 1 }] }" </pre> <ul style="list-style-type: none"> • 14: 배송장소 이동 시작 • 15: 배송장소 도착 완료 • 16: 서랍 열림 • 17: 서랍 닫힘 • 18: 서랍 잠금 • 19: 충전 시작 • 20: 충전 종료 • 21: 투숙객 이탈 • 22: 투숙객 이탈 후 재등록

From	To	인터페이스 항목	메시지 형식	비고
				<ul style="list-style-type: none">• 23: 투숙객 등록• 24: 배송 수령 완료• 25: 배송 수령 미완료• 26: 적재 감지• 27: 적재 미감지

From	To	인터페이스 항목	메시지 형식	비고
Robot GUI	RC	로봇 GUI 이벤트 전송 <code>/robot_gui/event</code>	<pre># RobotGuiEvent.msg int32 robot_id int32 rgui_event_id int32 task_id builtin_interfaces/Time timestamp string detail</pre>	<p>rgui_event_id</p> <ul style="list-style-type: none"> 100: [수령 완료] 클릭 101: 목적지 입력 완료 <ul style="list-style-type: none"> detail <ul style="list-style-type: none"> # 목적지 정보 "LOCATION_NAME" 102: 사용자 점유 상태 <ul style="list-style-type: none"> detail <ul style="list-style-type: none"> "OCCUPIED" or "VACANT" 103: [카드키로 입력] 선택 104: [서랍 열기] 클릭 105: [적재 완료] 클릭

From	To	인터페이스 항목	메시지 형식	비고
				<ul style="list-style-type: none"> 106: 인식모드 전환 요청 <ul style="list-style-type: none"> detail <ul style="list-style-type: none"> “0”: 대기모드 “1”: 등록모드 “2”: 추적모드 “3”: 엘리베이터모드


2.3 RC/AC <-> VS

From	To	프로토콜	인터페이스 항목	메시지 형식	비고



RC	VS	ROS2 Service	인식 모드 전환 요청 /vs/command/ set_vs_mode	<pre> # SetVSMode.srv # Request int32 robot_id int32 mode_id --- # Response int32 robot_id bool success </pre>	mode_id <ul style="list-style-type: none"> • 0: 대기모드 (후방 전용) • 1: 등록모드 (후방 전용) • 2: 추적모드 (후방 전용) • 3: 엘리베이터 외부 모드 (전방 전용) • 4: 엘리베이터 내부 모드 (전방 전용) • 5: 일반 주행모드 (전방 전용) • 6: 대기모드 (전방 전용)
----	----	--------------	--	---	---


RC	VS	ROS2 Service	<p>버튼 상태 감지 요청</p> <p>/vs/command/ button_status</p>	<pre># ButtonStatus.srv # Request int32 robot_id int32 button_id --- # Response int32 robot_id int32 button_id bool success float32 x float32 y float32 size bool is_pressed builtin_interfaces/Time timestamp</pre>	<p>x</p> <ul style="list-style-type: none"> • 0~1 <p>y</p> <ul style="list-style-type: none"> • 0~1 <p>size</p> <ul style="list-style-type: none"> • 0~1 <p>button_id</p> <ul style="list-style-type: none"> • 0: (현재 유일하게 감지되는 버튼) <ul style="list-style-type: none"> • 버튼이 2개 이상 감지될 경우 success=false • 1: 1층 • 2: 2층 • 3: 3층 • 4: 4층 • 5: 5층 • 6: 6층 • 7: 7층 • 8: 8층 • 9: 9층 • 10: 10층 • 11: 11층 • 12: 12층 • 13: B1층 • 14: B2층
----	----	--------------	--	---	---

					<ul style="list-style-type: none"> • 100: 하행버튼 • 101: 상행버튼 • 102: 열기버튼 • 103: 닫기버튼
RC	VS	ROS2 Service	엘리베이터 위치 및 방향 감지 요청 /vs/command/ elevator_status	<pre># ElevatorStatus.srv # Request int32 robot_id --- # Response int32 robot_id bool success int32 direction int32 position</pre>	<p>direction</p> <ul style="list-style-type: none"> • 0: upward • 1: downward <p>position</p> <ul style="list-style-type: none"> • 엘리베이터 현재 층

RC	VS	ROS2 Service	문 열림 감지 요청 /vs/command/ door_status	<pre># DoorStatus.srv # Request int32 robot_id --- # Response int32 robot_id bool success bool door_opened</pre>	door_opened <ul style="list-style-type: none">• 0: closed• 1: opened <div> 문 감지 여부로 판단</div>
----	----	--------------	---	---	---

RC	VS	ROS2 Service	현재 위치 감지 결과 <div>/vs/command/location</div>	<div><pre># Location.srv # Request int32 robot_id --- # Response int32 robot_id bool success int32 location_id</pre></div>	<table><tr><th>id</th><th>name</th></tr><tr><td>0</td><td>LOB_WAITING</td></tr><tr><td>1</td><td>LOB_CALL</td></tr><tr><td>2</td><td>RES_PICKUP</td></tr><tr><td>3</td><td>RES_CALL</td></tr><tr><td>4</td><td>SUP_PICKUP</td></tr><tr><td>5</td><td>ELE_1</td></tr><tr><td>6</td><td>ELE_2</td></tr><tr><td>101</td><td>ROOM_101</td></tr><tr><td>102</td><td>ROOM_102</td></tr><tr><td>201</td><td>ROOM_201</td></tr></table>	id	name	0	LOB_WAITING	1	LOB_CALL	2	RES_PICKUP	3	RES_CALL	4	SUP_PICKUP	5	ELE_1	6	ELE_2	101	ROOM_101	102	ROOM_102	201	ROOM_201
id	name																										
0	LOB_WAITING																										
1	LOB_CALL																										
2	RES_PICKUP																										
3	RES_CALL																										
4	SUP_PICKUP																										
5	ELE_1																										
6	ELE_2																										
101	ROOM_101																										
102	ROOM_102																										
201	ROOM_201																										

					<table><tr><th>id</th><th>name</th></tr><tr><td>202</td><td>ROOM_202</td></tr></table>	id	name	202	ROOM_202
id	name								
202	ROOM_202								
VS	RC	ROS2 Topic	장애물 감지 결과 <code>/vs/obstacle</code>	<pre># Obstacle.msg int32 robot_id bool dynamic float32 x float32 y float32 z</pre>	<div>dynamic<ul style="list-style-type: none">False: 정적 장애물True: 동적 장애물</div> <div><div> 감지될 때마다 발행</div></div> <div><div> 최종 출력 단위<ul style="list-style-type: none">X, Y, Z 좌표: 미터(m)거리: 미터(m)장애물 메시지: 미터(m)</div></div>				

VS	RC	ROS2 Topic	유리 문 상태 감지 결과 <code>/vs/glass_door_status</code>	<pre># GlassDoorStatus.msg int32 robot_id bool opened</pre>	<div> opened 산출 로직</div> <ul style="list-style-type: none">• '유리문' 감지됨: opened=False• 감지되지 않음: opened=True
----	----	------------	---	--	---

2.4 RC <-> IOC

From	To	인터페이스 항목	메시지 형식	설명
Service				
RC	IOC	서랍 잠금 제어 요청 <code>/ioc/control_lock</code>	<pre> # ControlLock.srv # Request int32 robot_id bool locked --- # Response int32 robot_id bool success </pre>	command: 수행할 명령 <ul style="list-style-type: none"> • true: 잠금 • false : 열림

From	To	인터페이스 항목	메시지 형식	설명
RC	IOC	서랍 열림 상태 확인 <code>/ioc/check_door_state</code>	<pre> # CheckDoorState.srv # Request int32 robot_id --- # ResponseGetItemState int32 robot_id bool is_opened </pre>	is_opened <ul style="list-style-type: none"> • true : 열린 상태 • false : 닫힌 상태

From	To	인터페이스 항목	메시지 형식	설명
RC	IOC	<p>물품 적재 여부 확인</p> <p><code>/ioc/check_item_loaded</code></p>	<pre># CheckItemLoaded.srv # Request int32 robot_id --- # Response int32 robot_id bool item_loaded</pre>	<p>item_loaded</p> <ul style="list-style-type: none"> • true : 물품이 감지된 상태 • false : 없는 상태

From	To	인터페이스 항목	메시지 형식	설명
Topic				
RC	IOC	로봇 상태 이벤트 (LED 반영용) <code>/roomie/status/robot_state</code>	<pre># RobotState.msg int32 robot_id int32 robot_state_id</pre>	robot_status_id <ul style="list-style-type: none"> • 0: 초기화 • 1: 충전상태 • 2: 작업대기 • 10: 픽업위치 이동 • 11: 픽업대기 • 12: 배송장소 이동 • 13: 수령대기 • 20: 호출위치 이동 • 21: 길안내 목적지 입력대기 • 22: 길안내 이동 • 23: 대상 탐색 • 30: 대기위치로 이동 • 31: 엘리베이터 탑승 • 90: 오류

From	To	인터페이스 항목	메시지 형식	설명
RC	IOC	카드 인식 요청 이벤트 <code>/ioc/read_card_request</code>	<pre># ReadCardRequest.msg int32 robot_id</pre>	
IOC	RC	카드 인식 응답 이벤트 <code>/ioc/read_card_response</code>	<pre># ReadCardResponse.msg int32 robot_id bool success int32 location_id</pre>	

2.5 RC <-> AGUI

From	To	프로토콜	인터페이스 항목	메시지 형식	
Topic					

From	To	프로토콜	인터페이스 항목	메시지 형식	
RC	AGUI	Topic	배터리 잔량 /roomie/status/ battery	<pre># BatteryStatus.msg int32 robot_id float32 charge_percentage bool is_charging</pre>	
RC	AGUI	Topic	현재 로봇 위치와 방향 /roomie/status/ roomie_pose	<pre># RoomiePose.msg int32 robot_id int32 floor geometry_msgs/Pose pose</pre>	

2.6 RC <-> AC

From	To	프로토콜	인터페이스 항목	메시지 형식	
RC	AC	Action	팔 회전 명령 <code>/arm/action/set_pose</code>	<pre> # SetPose.action # Goal int32 robot_id int32 pose_id --- # Result int32 robot_id bool success --- # Feedback (없음) </pre>	pose_id 0: init 초기자세 1: 준비자세 2: left 왼쪽 회전 자세 3: right 오른쪽 회전 자세 4: forward 전면 회전 자세 5: up 위쪽 회전 자세

From	To	프로토콜	인터페이스 항목	메시지 형식	
RC	AC	Action	버튼 클릭 명령 <code>/arm/action/click_button</code>	<pre> # ClickButton.action # Goal int32 robot_id int32 button_id --- # Result (RCS -> RMS 최종 결과) int32 robot_id bool success string message --- # Feedback (Arm(수행중) -> RC 진행 상황) int32 robot_id string status </pre>	status "MOVING_TO_TARGET", "ALIGNING_TO_TARGET", "PRESSING" "RETRACTING" "COMPLETED" "FAILED"

3 serial 통신

- AC <-> AD

3.1 AC <-> AD

3.1.1 기본 구조

- 형식: <명령코드:인자1,인자2,...> : < 와 종료 문자인 > 로 감싸서 메시지의 시작과 끝을 명확히 구분
- 구분자: 명령코드와 인자는 콜론(:), 인자 간에는 쉼표(,)로 구분

3.1.2 AC → AD

프로토콜	동기성	인터페이스 항목	메시지 형식	설명
Serial	동기	로봇팔 동작	<M,90,90,90,90>	M: 각도값 실행 <M:Joint1, Joint2, Joint3, Joint4>

3.1.3 AD → AC

프로토콜	동기성	인터페이스 항목	메시지 형식	
Serial	동기	동작 결과 응답	<OK>	1. 성공 시: <OK> 2. 실패 시: <ERR:101> a. <ERR:101> : 각도 범위 초과 b. <ERR:201> : 모터 과부하