

# 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
<h2>GUI 통신</h2>			
Admin GUI	RMS	WebSocket, HTTP	<ul style="list-style-type: none"><li>• 동기<ul style="list-style-type: none"><li>• 작업 목록 요청</li><li>• 시작 날짜</li><li>• 마지막 날짜</li><li>• 작업 유형</li><li>• 작업 상태</li><li>• 목적지</li></ul></li><li>• 특정 작업 상세정보 요청<ul style="list-style-type: none"><li>• 작업 ID</li></ul></li><li>• 로봇 목록 요청<ul style="list-style-type: none"><li>• 로봇 ID</li><li>• 모델명</li><li>• 로봇 상태</li><li>• 작업 상태</li></ul></li></ul>

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

# 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	<p><b>필터 미적용 시</b></p> <pre>{   "type": "request",   "action": "task_list",   "payload": {     "filters": {} // 빈 필터 = 전체 조회   } }</pre> <p><b>필터 적용 시</b></p> <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>	<p><b>task_type</b></p> <ul style="list-style-type: none"> <li>• 0: 음식배송</li> <li>• 1: 비품배송</li> <li>• 2: 호출</li> <li>• 3: 길안내</li> </ul> <p><b>task_status</b></p> <ul style="list-style-type: none"> <li>• 0: 접수됨</li> <li>• 1: 준비 완료</li> <li>• 2: 로봇 할당됨</li> <li>• 3: 픽업 장소로 이동</li> <li>• 4: 픽업 대기 중</li> <li>• 5: 배송 중</li> <li>• 6: 배송 도착</li> <li>• 7: 수령 완료</li> <li>• 10: 호출 이동 중</li> <li>• 11: 호출 도착</li> <li>• 20: 길안내 접수됨</li> <li>• 21: 길안내 중</li> <li>• 22: 길안내 도착</li> </ul>

HTTP					
API 명	HTTP 메서드	URL	요청 데이터 구조	응답 데이터 구조	비고
특정 작업 상세정보 조회	POST	/api/gui/task_detail	{           "type": "request",           "action": "task_detail",           "payload": {             "task_id": 12 //int           }         }	{           "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           }         }	

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

HTTP					
로봇 목록 조회	POST	/api/gui/ robot_list	<p><b>필터 미적용 시</b></p> <pre>{   "type": "request",   "action": "robot_list",   "payload": {     "filters": {} // 빈 필터 = 전체 조회   } }</pre> <p><b>필터 적용 시</b></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><b>location</b></p> <ul style="list-style-type: none"> <li>• 0: LOB_WAITING</li> <li>• 1: LOB_CALL</li> <li>• 2: RES_PICKUP</li> <li>• 3: RES_CALL</li> <li>• 4: SUP_PICKUP</li> <li>• 5: ELE_1</li> <li>• 6: ELE_2</li> <li>• 101: ROOM_101</li> <li>• 102: ROOM_102</li> <li>• 201: ROOM_201</li> <li>• 202: ROOM_202</li> </ul>

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_t ask	{           "type": "request",           "action": "create_call_task",           "payload": {             "location_name": "ROOM_201", // str             "task_type_id": 2 // intint, 0~3, 필수 (0: 음식, 1: 비품, 2: 호출, 3: 길안내)           }         }	{           "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           }         }	<b>location</b> <ul style="list-style-type: none"> <li>• 0: LOB_WAITING</li> <li>• 1: LOB_CALL</li> <li>• 2: RES_PICKUP</li> <li>• 3: RES_CALL</li> <li>• 4: SUP_PICKUP</li> <li>• 5: ELE_1</li> <li>• 6: ELE_2</li> <li>• 101: ROOM_101</li> <li>• 102: ROOM_102</li> <li>• 201: ROOM_201</li> <li>• 202: ROOM_202</li> </ul> <b>task_type</b> <ul style="list-style-type: none"> <li>• 0: 음식배송</li> <li>• 1: 비품배송</li> <li>• 2: 호출</li> <li>• 3: 길안내</li> </ul>

HTTP					
API 명	HTTP 메서드	URL	요청 데이터 구조	응답 데이터 구조	비고
호출 내역 조회	POST	/api/gui/get_call_history	{           "type": "request",           "action": "get_call_history",           "payload": {             "location_name": "ROOM_102" //str             "task_name": "TASK_006" //str           }         }	{           "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             }           }         }	

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

<b>HTTP</b>					
API 명	HTTP 메서드	URL	요청 데이터 구조	응답 데이터 구조	비고
음식 메뉴 조회	POST	/api/gui/get_food_menu	<pre>{   "type": "request",   "action": "get_food_menu",   "payload": {     "location_name": "ROOM_201" //string   } }</pre>	<pre>{   "type": "response",   "action": "get_food_menu",   "payload": {     "food_items": [ //array       {         "food_name": "스파게티", //string         "price": 15000, //int         "image": ".. (생략)" //       },       {         "food_name": "피자",         "price": 25000,         "image": ".. (생략)"       }     ]   } }</pre>	<p><b>Food</b></p> <ul style="list-style-type: none"> <li>• 0: 스파게티</li> <li>• 1: 피자</li> <li>• 2: 스테이크</li> <li>• 3: 버거</li> </ul> <p><b>Supply</b></p> <ul style="list-style-type: none"> <li>• 0: 칫솔</li> <li>• 1: 타월</li> <li>• 2: 생수</li> <li>• 3: 수저</li> </ul>

<b>HTTP</b>					
API 명	HTTP 메서드	URL	요청 데이터 구조	응답 데이터 구조	비고

HTTP					
API 명	HTTP 메서드	URL	요청 데이터 구조	응답 데이터 구조	비고
배송 작업 생성	POST	/api/gui/ create_delivery_task	{           "type": "request",           "action": "create_delivery_task",           "payload": {             "location_name": "ROOM_201", //str             "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	<pre>{   "type": "request",   "action": "get_task_list",   "payload": {     "location_name": "ROOM_201"   } }</pre>	<pre>{   "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"       }       ...     ]   } }</pre>	

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

<b>HTTP</b>					
주문 상세내역 조회	POST	/api/gui/get_order_history	<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": "음식배송", //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>	

<b>WebSocket</b>					
이벤트명	방향	URL	전송 데이터 구조	비고	

<b>WebSocket</b>				
호출 수락 알림	RMS → GGUI	URL	전송 데이터 구조	비고
		/api/gui/ws/guest/ {location_name}	{           "type": "event",           "action": "call_request_acceptance",           "payload": {             "task_name": "TASK_006", //str             "estimated_wait_time": 15 //int           }         }	

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

<b>WebSocket</b>				
이벤트명	방향	URL	전송 데이터 구조	비고

WebSocket				
방향	RMS → GGUI	URL	전송 데이터 구조	비고
배송 완료 알림		/api/gui/ws/guest/ {location_name}	{           "type": "event",           "action": "delivery_completion",           "payload": {             "task_name": "TASK_001", //str             "request_location": "ROOM_102" //str           }         }	

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

| |

## 1.3 RMS <-> SGUI

<b>HTTP</b>					
<b>API 명</b>	<b>HTTP 메서드</b>	<b>URL</b>	<b>요청 데이터 구조</b>	<b>응답 데이터 구조</b>	<b>비고</b>
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           }         }	

<b>WebSocket</b>				
<b>이벤트명</b>	<b>방향</b>	<b>URL</b>	<b>전송 데이터 구조</b>	<b>비고</b>

WebSocket				
음식 주문 발생 알림	RMS → SGUI	URL	전송 데이터 구조	비고
		/api/gui/ws/staff/{staff_id}	<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",           "quantity": 1, //int           "price": 15000 //int         }       ]     }   } }</pre>	<b>food</b> <ul style="list-style-type: none"> <li>• 0: 스파게티</li> <li>• 1: 피자</li> <li>• 2: 스테이크</li> <li>• 3: 버거</li> </ul>

WebSocket				
이벤트명	방향	URL	전송 데이터 구조	비고

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

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

<b>WebSocket</b>				
이벤트명	방향	URL	전송 데이터 구조	비고

WebSocket				
비품 요청 발생 알림	RMS → SGUI	/api/gui/ws/staff/{staff_id}	<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>	<b>supply</b> <ul style="list-style-type: none"> <li>• 0: 칫솔</li> <li>• 1: 타월</li> <li>• 2: 생수</li> <li>• 3: 수저</li> </ul>

WebSocket				
이벤트명	방향	URL	전송 데이터 구조	비고

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

○ ○ ○

## 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  /roomie/action/ perform_task	<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 -&gt; RMS 최종 결과) int32 robot_id int32 task_id bool success string message  ---  # Feedback (RCS(수행중) -&gt; RMS 진행 상황) int32 robot_id int32 task_id </pre>	<p><b>task_type_id</b></p> <ul style="list-style-type: none"> <li>• 0: 음식배송</li> <li>• 1: 비품배송</li> <li>• 2: 호출</li> <li>• 3: 길안내</li> </ul> <p><b>location_id</b></p> <ul style="list-style-type: none"> <li>• 0: LOB_WAITING</li> <li>• 1: LOB_CALL</li> <li>• 2: RES_PICKUP</li> <li>• 3: RES_CALL</li> <li>• 4: SUP_PICKUP</li> <li>• 5: ELE_1</li> <li>• 6: ELE_2</li> <li>• 101: ROOM_101</li> <li>• 102: ROOM_102</li> <li>• 201: ROOM_201</li> <li>• 202: ROOM_202</li> </ul> <p><b>order_info</b></p> <div style="background-color: #e0c0ff; padding: 10px;"> <span style="color: purple;">■</span> 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>	
<b>Service</b>					

From	To	프로토콜	인터페이스 항목	메시지 형식	
RC	RMS	Service	좌표 데이터 요청 <code>/roomie/command/get_locations</code>	<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>	
<b>Topic</b>					

From	To	프로토콜	인터페이스 항목	메시지 형식	
RC	RMS	Topic	로봇 상태 <code>/roomie/status/ robot_state</code>	<pre># RobotState.msg int32 robot_id int32 robot_state_id</pre>	<b>robot_state_id</b> <ul style="list-style-type: none"> <li>• 0: 초기화</li> <li>• 1: 충전상태</li> <li>• 2: 작업대기</li> <li>• 10: 픽업위치 이동</li> <li>• 11: 픽업대기</li> <li>• 12: 배송장소 이동</li> <li>• 13: 수령대기</li> <li>• 20: 호출위치 이동</li> <li>• 21: 길안내 목적지 입력대기</li> <li>• 22: 길안내 이동</li> <li>• 23: 대상 탐색</li> <li>• 30: 대기위치로 이동</li> <li>• 31: 엘리베이터 탑승</li> <li>• 90: 오류</li> </ul>
RC	RMS	Topic	배터리 잔량 <code>/roomie/status/ battery_status</code>	<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 <code>/roomie/status/ roomie_pose</code>	<pre># RoomiePose.msg int32 robot_id int32 floor_id geometry_msgs/Pose pose</pre>	<p><b>floor_id</b></p> <ul style="list-style-type: none"> <li>• 0: 1층</li> <li>• 1: 2층</li> <li>• 2: 3층</li> <li>• 3: 4층</li> </ul>

## 2.2 RC <-> RGUI

From	To	인터페이스 항목	메시지 형식	비고
<b>ROS2 Action</b>				
RC	RGUI	출발 카운트다운 <pre>/robot_gui/action/ start_countdown</pre>	<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>	<b>task_type_id</b> <ul style="list-style-type: none"> <li>• 0: 음식배송</li> <li>• 1: 비품배송</li> <li>• 2: 호출</li> <li>• 3: 길안내</li> </ul>

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

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><b>rgui_event_id</b></p> <ul style="list-style-type: none"> <li>• 1: 엘리베이터 버튼 조작 시작</li> <li>• 2: 엘리베이터 버튼 조작 종료</li> <li>• 3: 엘리베이터 탑승 시작</li> <li>• 4: 엘리베이터 탑승 종료</li> <li>• 5: 엘리베이터 하차 시작</li> <li>• 6: 엘리베이터 하차 종료</li> <li>• 7: 호출 이동 시작</li> <li>• 8: 호출 이동 종료</li> <li>• 9: 호출 번호 인식 완료 <ul style="list-style-type: none"> <li>• detail</li> </ul> </li> <li>• # 호출 번호 "101"</li> </ul> <ul style="list-style-type: none"> <li>• 10: 길안내 이동 시작</li> <li>• 11: 길안내 이동 종료</li> <li>• 12: 픽업장소 이동 시작</li> <li>• 13: 픽업장소 이동 종료 <ul style="list-style-type: none"> <li>• detail</li> </ul> </li> <li>• # 주문 정보</li> </ul>

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

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

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><b>rgui_event_id</b></p> <ul style="list-style-type: none"> <li>• 100: [수령 완료] 클릭</li> <li>• 101: 목적지 입력 완료 <ul style="list-style-type: none"> <li>• detail</li> </ul> </li> <li>• # 목적지 정보 "LOCATION_NAME"</li> </ul> <ul style="list-style-type: none"> <li>• 102: 사용자 점유 상태 <ul style="list-style-type: none"> <li>• detail</li> </ul> </li> <li>• "OCCUPIED" or "VACANT"</li> </ul> <ul style="list-style-type: none"> <li>• 103: [카드키로 입력] 선택</li> <li>• 104: [서랍 열기] 클릭</li> <li>• 105: [적재 완료] 클릭</li> </ul>

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

## 2.3 RC/AC <-> VS

From	To	프로토콜	인터페이스 항목	메시지 형식	비고

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

RC	VS	ROS2 Service	<p>버튼 상태 감지 요청</p> <pre>/vs/command/ button_status</pre>	<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><b>x</b></p> <ul style="list-style-type: none"> <li>• 0~1</li> </ul> <p><b>y</b></p> <ul style="list-style-type: none"> <li>• 0~1</li> </ul> <p><b>size</b></p> <ul style="list-style-type: none"> <li>• 0~1</li> </ul> <p><b>button_id</b></p> <ul style="list-style-type: none"> <li>• 0: (현재 유일하게 감지되는 버튼)       <ul style="list-style-type: none"> <li>• 버튼이 2개 이상 감지될 경우 success=false</li> </ul> </li> <li>• 1: 1층</li> <li>• 2: 2층</li> <li>• 3: 3층</li> <li>• 4: 4층</li> <li>• 5: 5층</li> <li>• 6: 6층</li> <li>• 7: 7층</li> <li>• 8: 8층</li> <li>• 9: 9층</li> <li>• 10: 10층</li> <li>• 11: 11층</li> <li>• 12: 12층</li> <li>• 13: B1층</li> <li>• 14: B2층</li> </ul>
----	----	--------------	--	---	---

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

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

RC	VS	ROS2 Service	현재 위치 감지 결과 <code>/vs/command/location</code>	<pre> # Location.srv  # Request int32 robot_id  ---  # Response int32 robot_id bool success int32 location_id </pre>	<table border="1"> <thead> <tr> <th><b>id</b></th><th><b>name</b></th></tr> </thead> <tbody> <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> </tbody> </table>	<b>id</b>	<b>name</b>	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
<b>id</b>	<b>name</b>																										
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 border="1"> <thead> <tr> <th><b>id</b></th><th><b>name</b></th></tr> </thead> <tbody> <tr> <td>202</td><td>ROOM_202</td></tr> </tbody> </table>	<b>id</b>	<b>name</b>	202	ROOM_202
<b>id</b>	<b>name</b>								
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>	<p><b>dynamic</b></p> <ul style="list-style-type: none"> <li>• False: 정적 장애물</li> <li>• True: 동적 장애물</li> </ul> <div style="border: 1px solid blue; padding: 10px;"> <span style="color: blue;">i</span>    감지될 때마다 발행     </div> <div style="border: 1px solid purple; padding: 10px;"> <span style="color: purple;">■</span>    <b>최종 출력 단위</b> <ul style="list-style-type: none"> <li>• X, Y, Z 좌표: 미터(m)</li> <li>• 거리: 미터(m)</li> <li>• 장애물 메시지: 미터(m)</li> </ul> </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>	<b>■ opened 산출 로직</b> <ul style="list-style-type: none"><li>• '유리문' 감지됨: opened=False</li><li>• 감지되지 않음: opened=True</li></ul>
----	----	------------	---	---	--

## 2.4 RC <-> IOC

From	To	인터페이스 항목	메시지 형식	설명
<b>Service</b>				
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>	<p><b>command: 수행할 명령</b></p> <ul style="list-style-type: none"> <li>• true: 잠금</li> <li>• false : 열림</li> </ul>

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

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

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

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	
		<b>Topic</b>			

From	To	프로토콜	인터페이스 항목	메시지 형식	
RC	AGUI	Topic	배터리 잔량 <code>/roomie/status/battery</code>	<pre># BatteryStatus.msg int32 robot_id float32 charge_percentage bool is_charging</pre>	
RC	AGUI	Topic	현재 로봇 위치와 방향 <code>/roomie/status/roomie_pose</code>	<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>	<b>pose_id</b> 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 -&gt; RMS 최종 결과) int32 robot_id bool success string message  ---  # Feedback (Arm(수행중) -&gt; RC 진행 상황) int32 robot_id string status </pre>	<b>status</b> "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>	<ol style="list-style-type: none"><li>1. 성공 시: &lt;OK&gt;</li><li>2. 실패 시: &lt;ERR:101&gt;<ol style="list-style-type: none"><li>a. &lt;ERR:101&gt; : 각도 범위 초과</li><li>b. &lt;ERR:201&gt; : 모터 과부하</li></ol></li></ol>