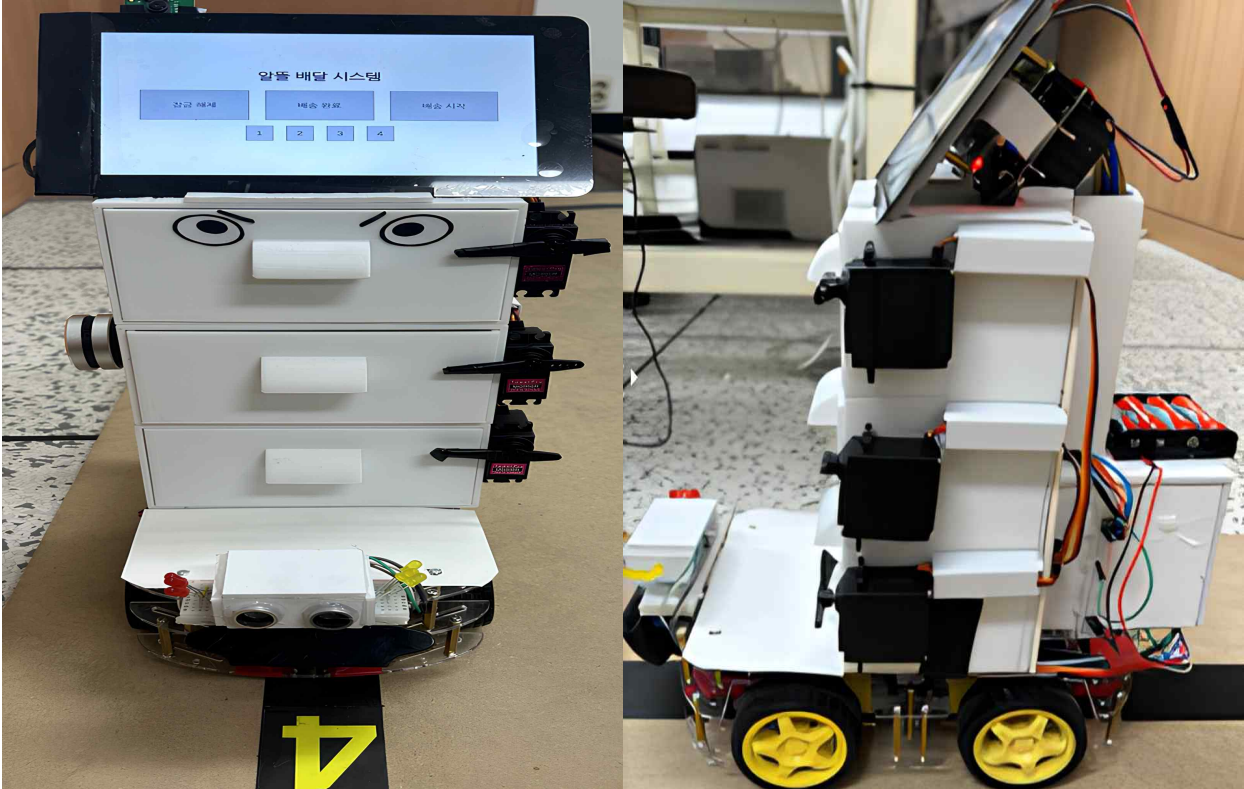


제20회 임베디드SW경진대회 개발완료보고서

[자유공모]

□ 개발 요약

팀 명	알뜰배달
	
작품명	QR인식을 통한 보안 강화 서빙 로봇
작품설명 (요약)	이 로봇은 QR 코드 인식을 통한 보관함 잠금 해제, 딥러닝을 이용한 숫자 인식 후 주문장소에 정지, OpenCV를 활용한 자율 주행 기능을 갖추어, 알뜰 배달 개념효율적이고 보안성 높은 서빙 및 배달을 목표로 제작
소스코드	https://github.com/kimjunzero/kimjunzero-2024ESWContest_free_1014.git
시연동영상	https://youtu.be/Ra99Bl9nL-w?si=9p7osuUxU7Cr-ovu

□ 개발 개요

○ 개발 작품 개요

- 현대 사회에서 비대면 서비스와 자동화된 시스템의 필요성은 점점 더 커지고 있습니다. 특히 코로나19 팬데믹 이후, 공공장소에서의 비대면 시스템 도입이 가속화되고 있습니다. 이러한 상황에서 알뜰배달의 개념을 자율주행 서빙 로봇에 적용하여, 공공장소에서 효율적으로 물품을 전달할 수 있는 시스템을 개발하고자 합니다. 이 로봇은 보안성을 강화하기 위해 적재함 잠금 및 숫자 인식 기능을 포함하여, 정확하고 안전한 자율주행 배달 시스템을 구현합니다.

○ 개발 목표

- 본 프로젝트의 목표는 공항, 학교 등 오픈된 공간에서 상품을 안전하게 배송하고, 소비자의 배달비용을 절감하는 효율적인 배달 시스템을 구축하는 것입니다. 특히, 배달의 민족의 '알뜰배달' 개념을 공공장소에 적용함으로써 여러 가게에서 로봇 시스템을 공동으로 활용해 보다 경제적이고 효율적인 배달 서비스를 제공할 수 있습니다.

○ 개발 작품의 필요성

- 공공장소에서 상품 도난 방지를 위한 보안 기술이 점점 중요해지고 있습니다. 로봇의 QR 코드 기반 잠금 기능은 이러한 보안 요구를 충족시켜줍니다. 예를 들어, 최근 다양한 장소에서 비대면 픽업 서비스가 늘어나면서, 물품이 중간에 도난당하는 사례가 발생하고 있습니다. 본 시스템은 이러한 문제를 해결하기 위해 설계되었으며, 사용자에게 안전하고 신뢰할 수 있는 비대면 상품 수령 방법을 제공합니다. 이와 더불어, 공공장소에서 중요한 문서나 고가의 물품을 안전하게 운반할 필요성이 증가함에 따라, 이러한 시스템의 필요성은 더욱 부각되고 있습니다.

○ 개발 작품의 활용성 및 기대효과

- 이 로봇 시스템은 단순한 상품 배달을 넘어 VIP 서비스를 제공할 수 있는 잠재력을 가지고 있습니다. 중요한 문서나 고가의 물품을 안전하게 전달할 수 있으며, QR 코드 인증을 통해 물품의 안전성을 확보하고, 허가된 사용자만 물품에 접근할 수 있도록 합니다. 이러한 시스템은 공공장소뿐만 아니라, 대형 이벤트, 박람회, 공항 등에서의 활용 가능성이 큽니다. 또한, 모바일 앱과의 연동을 통해 사용자는 실시간으로 로봇의 위치를 추적하고, 간편하게 배달 서비스를 이용할 수 있어, 사용자 경험을 크게 향상시킬 것입니다. 향후, 이 시스템은 다양한 비즈니스 모델에 적용되어 비대면 서비스의 표준이 될 가능성이 큽니다.

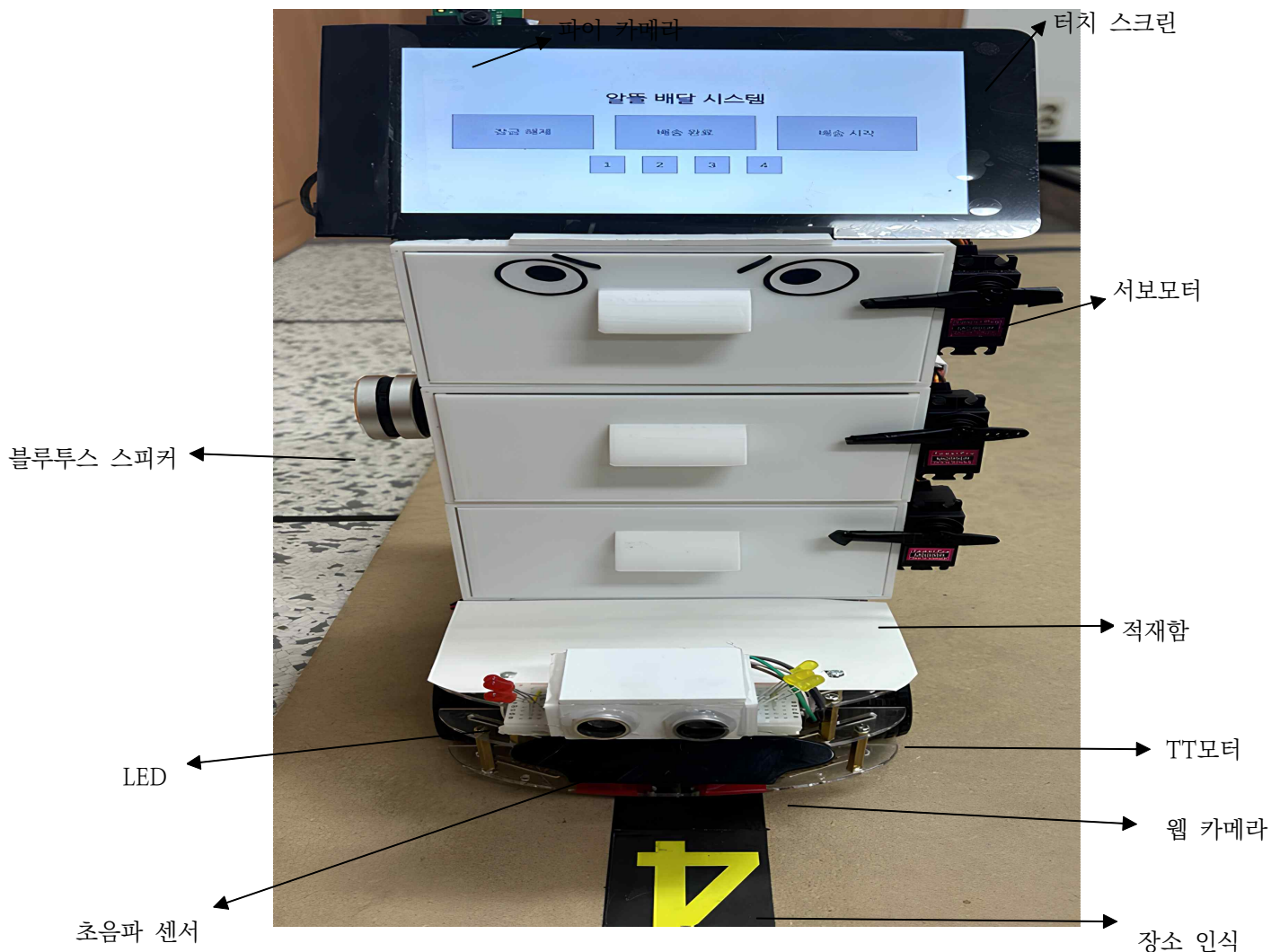
○ 기술적 트렌드와의 연관성

- 최근 AI와 로봇 공학의 발전으로 자율주행 기술이 크게 주목받고 있습니다. 이 프로젝트는 이러한 기술적 트렌드를 반영하여, 실시간 이미지 인식, 자율주행 알고리즘, IoT 기술 등을 결합한 종합적인 솔루션을 제공합니다. 특히, 공공장소에서의 비대면 서비스에 대한 요구가 증가하는 현 상황에서, 본 시스템은 안전하고 신뢰할 수 있는 대안으로 자리매김할 수 있습니다.

□ 개발 환경 설명

○ Hardware 구성

- 라즈베리파이4 : 로봇의 중앙 제어 장치로, QR코드 인식, 자율 주행 기능을 구현한다.
- 라즈베리파이 터치스크린 : 잠금해제, 배송 완료, 배송시작, 차선인식 화면 등 UI을 보여주기 위한 화면 기능
- TT모터 : TT모터를 활용하여 각 4개의 바퀴를 각각 제어한다. 이는 직선이나 회전차로에서 로봇이 효율적으로 주행을 하기 위함이다.
- L298N 모터 드라이버 : 앞바퀴, 뒷바퀴 총 2개의 바퀴
- 웹 카메라 : 차선인식을 하기 위한 카메라
- MG996GR 서보모터 : 잠금 기능을 담당하기 위해서 180도 회전한다.
- 파이 카메라 : QR코드를 인식하기 위한 카메라
- 적재함 : 상품을 담기 위한 총 3개의 적재함
- 블루투스 스피커 : 로봇의 음성 안내를 위한 스피커
- UPS 전원 공급기 : 선 없이 자율 주행을 하기 위한 전원 공급기
- 초음파 센서 : 돌발 상황 감지를 위한 센서
- LED : 돌발 상황 감지시 켜지는 LED 모듈



○ Hardware 기능 (제어 방법 등 서술)

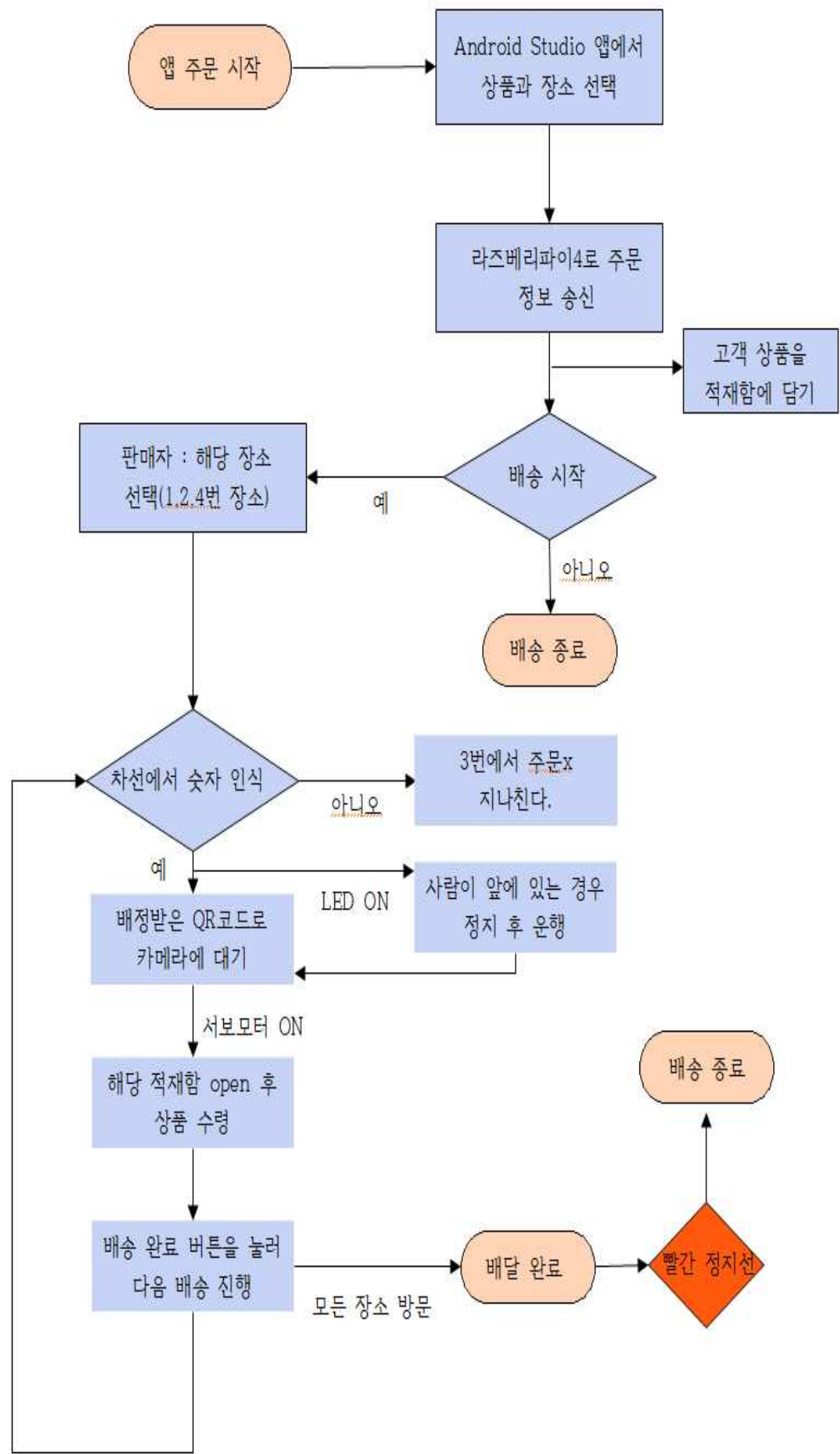
- 1. 라즈베리파이 터치스크린에서 주문 장소를 선택한다. 고객이 1번과 4번 장소에서 주문을 한 상황이면 1, 4번 클릭한다.
- 2. 배송 시작 버튼을 누르면 로봇이 배송을 시작. 4개의 모터 작동 ON
- 3. 주문한 장소가 아니면 로봇은 그냥 지나간다. 웹캠으로 해당 숫자를 인식하면 정지함과 동시에 해당장소에 도착했다고 알림
- 4. 고객이 앱으로 받은 QR코드를 파이 카메라에 가져다 댄다. 해당 적재함 해제
- 5. 고객이 상품을 수령하면, 고객이 배송완료 버튼을 클릭
- 6. 다음장소로 이동하여 수령 과정 반복!
- 7. 배송도중 사람이 있다면 초음파 센서가 작동하여 모터 제어 OFF, 사람이 사라지면 다시 배송
- 8. 다시 출발지로 돌아오게 된다.

○ Software 구성

- **알뜰 배달 APP 프론트엔드 개발** : 회원가입 및 로그인 화면 구현. 상품 목록 및 장바구니 화면 구현. QR code 및 주문 완료 화면 구현. recyclerView를 사용하여 상품 목록과 장바구니 상품 목록을 표시. Scroll View를 통해 스크롤 하여 상품 확인. 선택한 상품의 가격을 총 가격, 배송료, 할인금액, 총 결제 금액 등이 표시되어 사용자에게 주문 내역을 명확히 전달, CheckBox를 통해 주문할 배송 위치를 선택
- **알뜰 배달 APP 백엔드 개발** : 외부 라이브러리 Firebase와 연동하여 사용자들의 회원가입 및 로그인 관리(UID, 이메일, 비밀번호). 클릭 및 롱 클릭 이벤트를 처리하여 클릭 이벤트가 발생하면 해당 이벤트를 처리 후 토스트 메시지에 표시. 클릭된 이미지를 장바구니에 추가하고 장바구니 액티비티로 전환. 장바구니에 있는 아이템을 표시하기 위해 RecyclerView의 어댑터를 구현. 리스트에 저장된 장바구니 아이템을 사용하여 RecyclerView에 데이터를 제공 후 RecyclerView에 표시할 장바구니 아이템들의 데이터를 관리하고, 해당 데이터를 RecyclerView에 바인딩하여 사용자에게 표시. 장바구니 화면을 표시하고, 사용자가 선택한 상품을 서버에 전송. 체크박스 및 버튼 등의 사용자 인터페이스 요소에 대한 이벤트 리스너를 설정. HTTP POST 요청을 보내고, 서버에서 응답을 받은 후 QR코드를 생성하여 새로운 화면으로 이동. 선택된 상품의 이름을 TextView에 업데이트 및 가격 계산. 바이트 배열을 디코딩하여 QR코드를 표시.

○ Software 설계도 (흐름도 및 클래스 다이어그램 등 / 개발언어에 따라 선택)

알뜰 배달 흐름도



○ Software 기능 (필요 시 알고리즘 설명 포함)

- Altteul Deliver App :

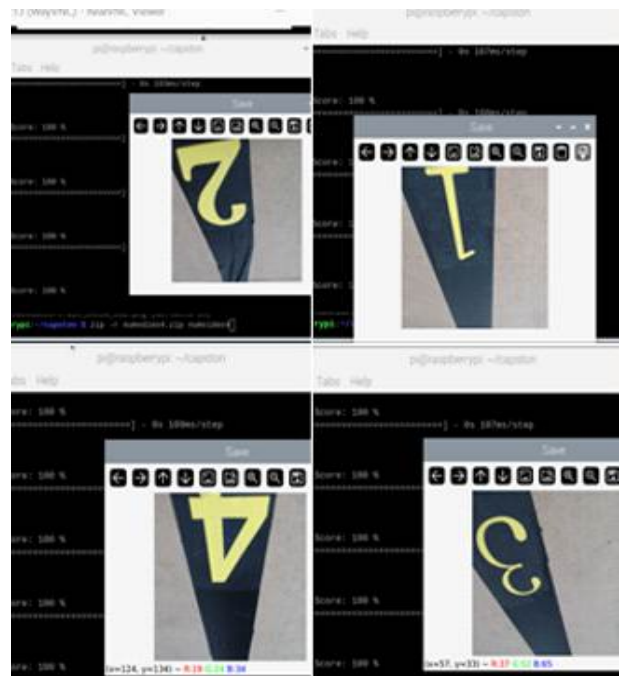
1. 회원 관리 및 상품 주문: Firebase를 통해 회원가입 및 로그인 기능을 제공하고, 상품 목록 조회, 장바구니 관리, 결제 정보 확인, QR 코드를 통한 주문 확인 및 완료 기능을 포함합니다.
2. 장바구니 및 결제 처리: 선택한 상품을 장바구니에 추가하고, 최종 결제 금액을 계산하여 사용자에게 보여줍니다. 서버와의 연동을 통해 주문 데이터를 처리합니다.

- 자율주행 로봇 :

1. 이미지 인식 및 주행 제어: OpenCV를 사용해 실시간으로 이미지 처리(회색화, 블러링, 임계값 처리)를 수행하고, 라즈베리파이로 주행 경로를 제어합니다. 숫자 인식을 통해 특정 에서 멈추고, 상품을 배송합니다.



차선 인식 후 이미지 처리



딥러닝으로 숫자 인식을 높이기

2. QR 코드 및 음성 안내: QR 코드를 인식해 잠금을 해제하고, 음성으로 도착 안내 및 배송 완료 확인을 합니다.

○ 프로그램 사용법 (Interface)

- Atteul Deliver APP :

1. 회원가입/로그인: 앱을 실행하고, 회원가입 또는 로그인 버튼을 클릭하여 Firebase를 통해 계정을 관리합니다.
2. 상품 선택 및 결제: 상품을 선택해 장바구니에 추가하고, 결제 화면에서 총 결제 금액을 확인한 후 주문을 완료합니다.
3. QR 코드 사용: 주문 완료 후 생성된 QR 코드를 로봇에 스캔하여 상품을 수령합니다.



- 자율주행 로봇 :

1. 주행 시작: 로봇의 터치스크린에서 '배송시작' 버튼을 터치하여 주행을 시작합니다.
2. 상품 수령: 로봇이 도착하면, QR 코드를 스캔해 적재함을 열고 상품을 수령합니다. '배송 완료' 버튼을 눌러 다음 배송을 진행합니다.

알뜰 배달 시스템



1, 2, 3장소에 주문이 들어왔을 경우

○ 개발환경 (언어, Tool, 사용시스템 등)

▶ 라즈베리파이 4:

- 언어 : python3
- Tool : openCV, TensorFlow, keras
- 사용 시스템 : 라즈베리파이 4는 자율주행 로봇의 주행 제어와 이미지 처리 작업을 담당합니다. Python 3을 사용하여 OpenCV를 통해 이미지 처리(회색화, 블러링, 임계값 처리)를 구현하고, TensorFlow와 Keras를 통해 숫자 인식을 수행합니다. 라즈베리파이에 연결된 터치스크린을 통해 사용자 인터페이스를 제공하며, QR 코드 인식과 서보모터 제어를 통한 적재함 잠금 해제 기능도 포함되어 있습니다.

▶ Android Studio:

- 언어 : Java, kotlin
- Tool : Android SDK, Firebase, XML, RecyclerView
- 사용 시스템 : Android Studio는 Altteul Delivery APP의 개발 환경입니다. Java와 Kotlin을 주 언어로 사용하여 앱의 프론트엔드와 백엔드를 구현합니다. Firebase를 연동하여 사용자 인증 및 데이터 관리를 수행하며, XML을 통해 앱의 사용자 인터페이스(UI)를 구성합니다. RecyclerView를 사용하여 상품 목록과 장바구니 항목을 효율적으로 관리하고, ScrollView와 CheckBox를 통해 사용자가 주문을 쉽게 확인하고 결제할 수 있도록 합니다. 앱은 또한 QR 코드 생성을 통해 주문 완료 및 확인 기능을 제공합니다.

□ 개발 프로그램 설명

○ 함수별 기능

▶ **drive.py**

1. `motor_setup()`: 모터 제어를 위한 GPIO 핀과 PWM을 초기화합니다. 모터의 방향과 속도를 제어할 수 있도록 설정합니다.
2. `motor_control(front_left_speed, front_right_speed, rear_left_speed, rear_right_speed)`: 각 모터의 속도와 방향을 제어합니다. 입력된 속도 값에 따라 PWM 듀티 사이클과 모터의 방향을 설정합니다.
3. `detect_red(frame)`: 입력된 영상에서 빨간색 영역의 픽셀 수를 감지하여 반환합니다. 빨간색 마스크를 생성하여 빨간색의 존재를 확인합니다.
4. `detect_yellow(frame)`: 입력된 영상에서 노란색 영역의 픽셀 수를 감지하여 반환합니다. 노란색 마스크를 생성하여 노란색의 존재를 확인합니다.
5. `main()`: 프로그램의 주요 실행 루프를 포함합니다. 카메라로부터 영상을 캡처하고, 차선 인식을 위해 이미지 처리를 수행하며, 모터를 제어하여 로봇을 자율적으로 주행시킵니다. 감지된 색상에 따라 로봇의 동작을 제어하고, 처리된 이미지를 소켓을 통해 전송합니다.

▶ **index.html**

- 이 HTML 파일은 “알뜰 배달 시스템” 웹 페이지를 구성합니다. Flexbox를 사용해 중앙에 정렬된 버튼과 비디오 스트림을 표시하며, Socket.IO를 통해 서버와 실시간 통신합니다.
1. 스타일 및 레이아웃 : 중앙에 배치된 버튼과 제목을 포함합니다.
 2. Socket.IO 통신 : 서버와 연결해 버튼의 표시 상태와 비디오 스트림을 제어합니다.
 3. 버튼 기능 : “잠금 해제” > 카메라 잠금 해제 요청을 서버로 보냅니다.
“배송 시작” > 배송 시작 요청을 서버로 보냅니다.
 4. 비디오 스트림 : 서버 상태에 따라 스트림을 표시합니다.

▶ **line.py**

1. `canny_edge_detection(img)`:
입력 이미지에서 Canny 엣지 검출을 수행합니다.
GaussianBlur를 사용하여 노이즈를 줄인 후, Canny 함수로 엣지를 검출합니다.

2. `region_of_interest(img, vertices):`

이미지에서 관심 영역(Region of Interest, ROI)을 지정하고, 해당 영역만을 남깁니다.
vertices로 정의된 다각형 영역을 마스크로 설정하여 그 외의 영역을 제거합니다.

3. `draw_lines(img, lines):`

검출된 차선을 이미지에 그립니다.

cv2.line을 사용하여 각 차선을 초록색 선으로 그립니다.

4. `main():`

프로그램의 메인 함수로, 실시간으로 웹캠에서 영상을 받아 차선을 검출합니다.

ROI 설정: 관심 영역을 설정하여 차선 검출을 집중시킵니다.

루프: 매 프레임마다 차선 검출을 수행하고, 검출된 차선을 원본 이미지에 표시합니다.

HoughLinesP: Hough 변환을 사용해 이미지에서 직선을 검출합니다.

▶ **opencv_qr.py**

1. `set_servo_degree(pin, degree)`

- 서보모터를 제어하여 특정 각도로 회전시킵니다.
- 입력: 서보모터가 연결된 GPIO 핀 번호(pin)와 회전시킬 각도(degree).
- 기능: 각도를 듀티 사이클로 변환하여 해당 핀의 서보모터를 제어합니다. 각도는 0도에서 180도 사이로 제한됩니다.

2. `unlock_and_lock_drawer(drawer)`

- 지정된 서랍을 잠금 해제하고 다시 잠급니다.
- 입력: 서랍의 이름(drawer, 예: 'top_drawer').
- 기능: 서랍에 해당하는 서보모터를 180도로 회전시켜 잠금 해제를 수행합니다.

3. `detect_and_display_qr_code(image)`

- QR 코드를 감지하고 그 내용을 화면에 표시합니다.
- 입력: 카메라로부터 캡처된 이미지(image).
- 기능: 이미지에서 QR 코드를 감지하고, 인식된 데이터가 있으면 해당 데이터를 이미지에 표시합니다. QR 코드의 위치를 사각형으로 표시하며, 감지된 데이터를 반환합니다.
- 출력: QR 코드 데이터 또는 False (QR 코드가 없을 경우).

4. `show_custom_message(qrnumber)`

- QR 코드로 인식된 번호에 따라 사용자에게 서랍이 열렸음을 알리는 메시지를 표시합니다.
- 입력: QR 코드에서 추출된 서랍 번호(qrnumber).
- 기능: 간단한 Tkinter GUI 창을 통해 사용자에게 알림을 표시합니다. 알림 창은 특정 시간 이 지나면 자동으로 닫힙니다.

5. `main()`

프로그램의 메인 루프를 실행하며, QR 코드를 인식하여 서랍을 제어합니다.

기능: 1. Picamera2를 사용하여 실시간으로 이미지를 캡처합니다.

2. QR 코드를 인식하고, 인식된 데이터가 1, 2, 3 중 하나일 경우 해당 서랍을 잠금 해제합니다.
3. 서랍이 열리면 사용자에게 알림을 표시하고, 카메라를 정지하며 프로그램을 종료합니다.
4. 'q' 키를 누르면 프로그램을 중지하고 모든 리소스를 정리합니다.

▶ server.py

서버 초기화: Flask와 SocketIO를 설정하고, ZMQ 소켓을 통해 비디오 스트림을 수신합니다.

웹 인터페이스: home() 함수가 index.html 페이지를 제공합니다.

비디오 스트리밍: video_feed() 경로에서 generate_frame() 함수를 사용하여 실시간 비디오 스트리밍을 제공합니다.

소켓 통신: 클라이언트의 소켓 연결을 확인하고, 이벤트를 통해 버튼의 표시와 비디오 스트리밍 상태를 업데이트합니다.

QR 코드 및 서브프로세스 제어: /unlock 경로에서 QR 코드 인식 프로그램을 실행합니다.

/data 경로에서 QR 코드와 관련된 데이터를 처리하고 QR 코드를 생성합니다. /start 경로에서 배송을 시작하고 관련 상태를 업데이트합니다.

이 애플리케이션은 비디오 스트리밍, QR 코드 인식, 서보모터 제어와 같은 기능을 포함하여 웹 기반의 인터페이스를 제공합니다.

○ 기술적 차별성

- QR 코드 기반 잠금 해제: 사용자별로 고유한 QR 코드를 생성해 상품 수령 시 적재함의 잠금을 해제합니다. 이는 배송의 안전성을 높이며, 각 고객이 고유한 방식으로 상품을 수령할 수 있게 합니다.
- 자율주행 로봇의 이미지 인식: OpenCV와 TensorFlow를 활용하여 자율주행 중 실시간으로 이미지를 처리하고, 특정 숫자나 색상을 인식하여 정확한 위치에 멈추거나 주행 방향을 결정합니다. 이를 통해 로봇이 복잡한 환경에서도 안정적인 배송을 수행할 수 있습니다.

□ 개발 중 발생한 장애요인과 해결방안

○ 모터 제어 구동

총 4개의 모터를 2개의 L298N 모터 드라이버를 사용하여 제어하였습니다. 앞바퀴 2개를 통해 잘 동작하는 것을 확인했으며, 뒷바퀴도 정상적으로 돌아가는 것을 확인하여 모든 4개의 바퀴가 잘 굴러갔습니다. 그러나 OpenCV를 활용하여 차선을 따라 주행하는 과정에서 원하는 방향으로 가지 않아서 구동 제어에 문제가 있음을 느꼈습니다. 이에 따라 모든 진행 상황을 멈추고 모터 제어에 집중했습니다.

먼저, 다음과 같은 문제를 점검했습니다:

1. 모터 고장
2. 모터 드라이브 고장

3. 코드상의 문제

문제 상황을 좁히기 위해 하나씩 모든 부분을 교체하고 수정하며 확인했으나, 여전히 모터 제어 테스트가 제대로 작동하지 않았습니다. 로봇의 무게가 무거워서 최대 출력의 한계가 있을 수 있다고 생각했으나, 모든 모터가 돌아갔으므로 이를 제외했습니다. 정확한 상황은 모든 모터를 전진 방향으로 돌렸을 때 1~2개의 바퀴가 뒤로 돌았고, 후진 방향으로 했을 때는 일부 바퀴가 전진 방향으로 회전하여 어지러운 상황이었습니다.

임베디드 시스템 설계 과목에서 점프선을 브레드보드에 연결할 때 잘 구동되지 않았던 경험이 있어, 점프선을 교체하자마자 문제가 해결된 사례를 기억했습니다. 이번에도 같은 문제가 있을 수 있다고 생각하여, 모터선을 하나하나 정리하며 새롭게 선을 교체하였더니 정상적으로 모터 제어가 되었습니다. 또한, TT 모터를 사용했는데, 속도 위주의 토크를 사용했으나 로봇의 무게로 인해 주행이 잘 되지 않았습니다. 이를 개선하기 위해 힘 토크를 사용하는 방향으로 변경하여 주행이 잘 되도록 조정했습니다.

모터 제어 문제는 해결하였으나, 차선 인식에 따른 모터 제어가 제대로 이루어지지 않았습니다. 차선을 따라 로봇의 구동이 제대로 제어되지 않는 문제가 있었습니다.

○ Android Studio와 라즈베리파이4 간의 서버 연동 및 데이터 송신 문제

- 개발 과정에서 여러 가지 장애 요인이 발생했으며, 각 문제에 대해 구체적인 해결 방안을 마련했습니다. 첫째, 안드로이드 앱과 라즈베리파이 서버 간의 데이터 전송 지연 및 손실 문제가 있었습니다. 특히 비디오 스트리밍과 QR 코드 데이터 전송 시 네트워크 불안정성으로 인해 데이터가 원활하게 전송되지 않았습니다. 이를 해결하기 위해 비동기 데이터 전송과 오류 재전송 메커니즘을 구현했습니다. ZeroMQ를 사용하여 비동기 통신을 설정하고, 데이터 전송의 신뢰성을 높였습니다.
- 라즈베리파이의 처리 능력 부족으로 인해 비디오 스트리밍과 QR 코드 인식 작업에서 성능 저하가 발생했습니다. 라즈베리파이의 CPU와 GPU 자원을 효율적으로 분산 사용하도록 최적화하였고, 영상 해상도를 낮추어 처리 부담을 줄였습니다. 성능이 더 높은 Raspberry Pi 4를 사용하는 방법도 고려하여 하드웨어 성능 문제를 해결했습니다.
- 웹 서버와 클라이언트 간 실시간 데이터 동기화에서 지연 및 불일치가 발생했습니다. SocketIO와 ZeroMQ의 설정을 조정하여 데이터 전송 속도와 신뢰성을 개선했으며, 소켓 연결의 버퍼링 및 대기 시간을 최적화하여 실시간 데이터를 보다 정확하게 전달할 수 있도록 했습니다. 클라이언트와 서버 간의 통신을 검증하고 로깅을 통해 문제가 발생한 경우 신속하게 원인을 파악하고 조치했습니다.
- QR 코드 인식의 정확성 문제가 있었습니다. 다양한 환경 조건에서 QR 코드 인식이 부정확한 경우가 많았는데, 이를 해결하기 위해 이미지 전처리 작업을 수행했습니다. 밝기와 대비 조정을 통해 QR 코드의 가독성을 향상시켰고, 카메라의 해상도와 초점을 조정하여 인식 품질을 개선했습니다. 이를 통해 다양한 조명 조건에서도 QR 코드 인식의 정확성을 높일 수 있었습니다. 이러한 해결 방안들을 통해 시스템의 신뢰성과 효율성을 크게 향상시킬 수 있었습니다.

□ 개발결과물의 차별성

○ 실시간 비디오 스트리밍 및 제어 통합

- 기존 유사작품: 대부분의 기존 유사 시스템은 비디오 스트리밍과 제어 기능이 분리되어 있거나, 기본적인 기능만 제공하며 별도의 하드웨어나 소프트웨어가 필요합니다.
- 차별성 및 우수성: 본 시스템은 Flask와 SocketIO를 활용하여 웹 인터페이스에서 실시간으로 비디오 스트리밍을 제공하며, 동시에 서버모터 제어와 QR 코드 인식을 통합합니다. 이를 통해 사용자는 웹 기반의 직관적인 인터페이스를 통해 실시간으로 비디오를 모니터링하고 제어할 수 있어 높은 편리성을 제공합니다.

○ QR 코드 기반 자동화 및 주문 관리

- 기존 유사작품: 많은 유사 시스템은 QR 코드 스캔 기능을 단순히 정보 수집이나 인증 목적으로만 사용합니다. 이들 시스템은 QR 코드 인식을 통해 기본적인 정보를 수집하거나, 시스템 작동에 필요한 단일 기능만을 지원합니다.
- 차별성 및 우수성: 본 시스템은 QR 코드 인식 기능을 통해 주문 관리와 서랍 잠금 해제까지 자동화하였습니다. QR 코드로 주문 정보를 수집하고, 해당 주문에 맞는 서랍을 자동으로 잠금 해제하는 기능을 구현하여, 복잡한 주문 처리를 간소화하고 자동화합니다. 이로 인해 사용자 경험을 극대화하고 효율성을 높였습니다.

○ 실시간 상태 업데이트 및 사용자 알림

- 기존 유사작품: 기존 시스템은 비디오 스트리밍과 제어 기능이 통합된 경우에도, 실시간 상태 업데이트나 사용자 알림 기능이 부족하거나 별도로 구현해야 하는 경우가 많습니다.
- 차별성 및 우수성: 본 시스템은 소켓 통신을 통해 비디오 스트리밍 상태와 버튼 상태를 실시간으로 업데이트하며, 사용자에게 주문 처리 상태 및 시스템 알림을 즉각적으로 전달합니다. 이러한 실시간 피드백과 알림 기능은 사용자와 시스템 간의 상호작용을 강화하며, 시스템의 운영 효율성과 사용 편의성을 높입니다.

○ 웹 기반의 직관적인 사용자 인터페이스

- 차별점: 본 시스템은 Flask와 SocketIO를 활용하여 웹 기반의 사용자 인터페이스를 제공합니다. 웹 페이지를 통해 비디오 피드를 실시간으로 스트리밍하고, QR 코드 인식 결과를 사용자에게 보여줍니다.

우수성: 사용자는 별도의 애플리케이션 없이 웹 브라우저만으로 시스템을 제어하고 모니터링할 수 있습니다. 이는 시스템 접근성을 높이며, 사용자가 편리하게 조작할 수 있도록 합니다.

□ 개발결과물의 파급력 및 기대효과

○ 다양한 산업 분야에서의 활용

- 내용: 본 시스템은 물류 창고에서의 실시간 비디오 모니터링, 스마트 오피스에서의 자동화된 자산 관리, 보안 시스템에서의 비디오 감시 등 여러 분야에 적용 가능합니다. QR 코드 인식 및 비디오 스트리밍 기능이 통합되어 있어, 다양한 산업에서 수요가 높습니다.

- 기대효과: 다양한 분야에 적용 가능하여 여러 산업에서 동시에 채택될 수 있으며, 이는 시스템의 시장성을 크게 높이고, 경쟁력 있는 솔루션으로 자리잡을 수 있게 합니다.

○ 비용 효율적인 솔루션

- 내용: 기존의 고가 하드웨어 솔루션 대신, 웹 기반의 소프트웨어와 저비용 하드웨어를 활용하여 비용을 절감했습니다. 특히, 비디오 스트리밍과 QR 코드 인식을 통합하여 시스템 설치 및 유지보수 비용을 최소화할 수 있습니다.

- 기대효과: 적은 비용으로 높은 기능성을 제공함으로써, 비용 효율성을 중시하는 고객에게 매력적인 선택이 될 수 있으며, 시장에서 가격 경쟁력을 강화합니다.

○ 실시간 비디오 스트리밍과 QR 코드 인식 통합

- 내용: 실시간 비디오 스트리밍을 통해 사용자는 현재 상황을 즉시 확인할 수 있으며, QR 코드 인식을 통해 자동으로 작업을 수행할 수 있습니다. 이 두 기능이 통합되어 사용자는 보다 신속하게 작업을 처리할 수 있습니다.

- 기대효과: 업무의 효율성을 크게 높이고, 작업 처리 속도를 향상시켜 사용자의 편의성과 생산성을 증대시킵니다.

○ 웹 기반의 직관적인 사용자 인터페이스

- 내용: 사용자 인터페이스는 웹 브라우저를 통해 접근 가능하며, 별도의 소프트웨어 설치가 필요 없습니다. 이는 시스템 설정과 운영이 매우 간편하게 이루어질 수 있도록 합니다.

- 기대효과: 사용자는 복잡한 소프트웨어 설치나 학습 없이 바로 시스템을 활용할 수 있어, 사용자 경험을 크게 향상시키고, 빠른 시스템 도입과 적응을 가능하게 합니다.

□ 개발 일정

No	내용	2022年															
		7月				8月				9月				10月			
1	바퀴 모터 제어																
2	기능 구현 및 개발																
3	시험 평가 및 테스트																

□ 팀 업무 분장

No	구분	성명	참여인원의 업무 분장
1	팀장	김준영	<p>자동차의 바퀴 모터 제어와 차선 인식 시스템을 담당. 초음파 센서 등 하드웨어 부분 관여</p> <ul style="list-style-type: none"> - 모터 제어 및 pwm 설정 - 차선 인식 알고리즘 개발 - 초음파 센서를 활용한 주행 기능 - 블루투스 스피커 모듈
2	팀원	김세원	<p>시스템의 데이터 처리와 사용자 인터페이스를 담당하며, 서버 구축을 만들며 전체적으로 소프트웨어 담당.</p> <ul style="list-style-type: none"> - Flask 기반 서버 구현 - 실시간 비디오 스트리밍 - QR 코드 인식 기능 - 데이터 전송 및 통신 - 숫자 인식 딥러닝 학습
3	팀원	김승환	<p>하드웨어 관련 작업을 담당하며, 적재함 3D프린팅으로 제작, 서보모터 제어, 하드웨어 부분 관여</p> <ul style="list-style-type: none"> - 서보모터 설정 및 회로도 구현 - 3D프린팅 외관 구현 - 모터 제어 및 pwm 설정 - 숫자 인식 딥러닝 학습