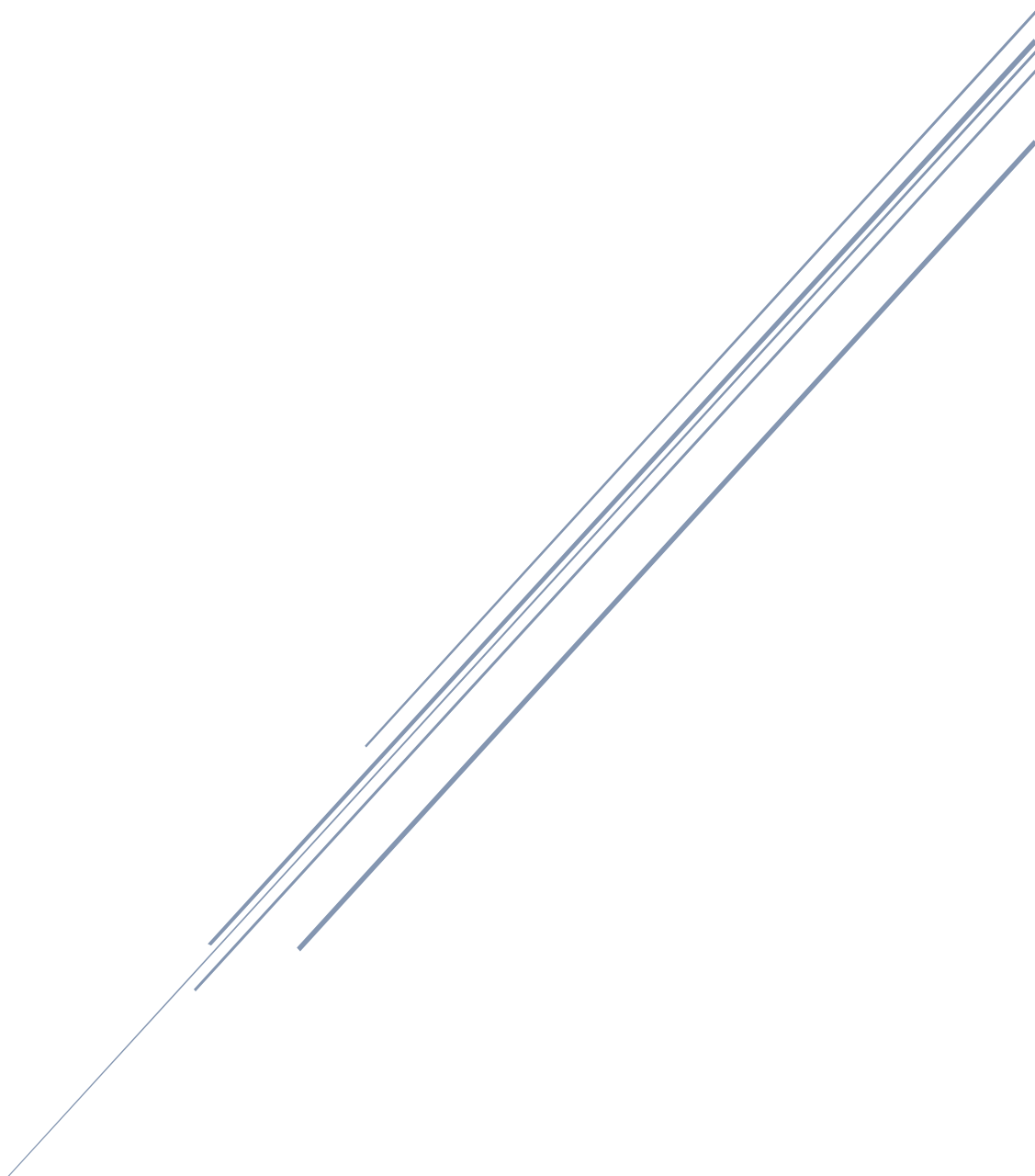


# 스마트 AIOT 길안내 시스템

[실버로드]



송이건 우승민 유승경 장대훈 조우찬

## 목 차

1. 작품 개요.....	2
1-1 개발 배경.....	2
1-2 작품 목표.....	3
2. 작품 설명.....	4
2-1 주요 동작 및 특징.....	4
2-2 전체 시스템 구성.....	5
2-3 개발 환경.....	10
3. 단계별 제작 과정.....	14
4. 사용한 제품 리스트.....	23
5. 추후 개발 요소.....	24
6. 참고문헌.....	25

## 1. 작품 개요

### 1-1 개발 배경

현재 우리나라는 초고령 사회로 진입하고 있으며, 장애인의 사회 참여와 공공시설 이용 또한 증가하고 있습니다. 하지만, 현실적으로 노약자와 장애인은 복잡한 실내 공간(병원, 공공기관 등)에서 목적지를 찾는데 많은 불편을 겪고 있습니다.

실제로 통계청에 따르면, 우리나라의 65세 이상 고령 인구 비율은 2023년 18.4%에서 2025년 20.6%로 증가해 초고령사회에 진입할 것으로 예측됩니다. 또한, 장애인 등록 인구는 약 270만 명을 넘어섰습니다. <그림 1>

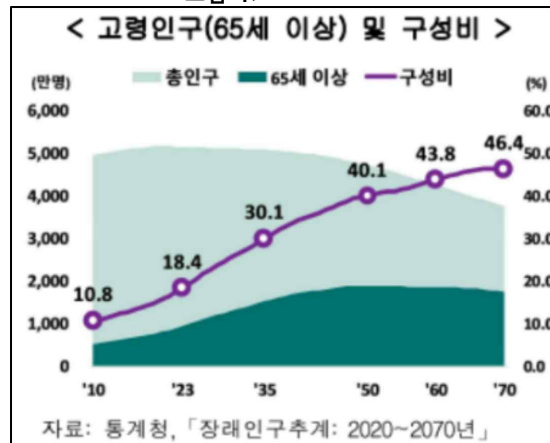


그림 고령인구 65세 이상 구성비

또한, 2021디지털정보격차실태조사에 따르면 고령층의 모바일 기기 이용능력에서 '필요한 앱 설치 및 이용' 부분이 60대는 55.8%, 70대 이상은 20.9%에 불과합니다. 그림 2



그림 2021 디지털정보격차실태

이러한 배경에서 교통 약자가 공공시설 내에서 자율적으로 이동할 수 있는 시스템의 필요성이 커지고 있지만, 일반인을 기준으로 설계되어 있어, 장애 유형이나 신체 조건에 따른 안내 최적화가 부족한 실정입니다.

이에 우리 팀은 누구나 목적지까지 휴대폰이나 어플 사용하지 않고도 안전하고 정확하게 이동할 수 있는 실내 길안내 시스템을 구상하게 되었습니다. 또한, 아이디어를 모으면서 몸이 불편하신 분이나 환자도 이동할 수 있는 것이 필요하다고 생각하여 로봇으로도 이송할 수 있는 시스템을 개발했습니다.

## 1-2 작품 목표

주요 목표는 스마트 길안내 시스템은 초고령사회 진입으로 인한 다수의 노약자와 장애인이 병원 내에서 편리하게 길안내를 목표로 합니다. 이를 통해 이동 불편 문제를 해소하고, 정보 격차로 인해 모바일 기기 사용이 어려운 노약자와 장애인도 쉽게 목적지에 도달할 수 있는 환경을 구축하는 것입니다. 현재 대학병원 등 대형병원에서는 바닥에 각 병실 및 진료실의 위치가 적혀 있음에도 매우 복잡한 구조와 수많은 인파로 인하여 여전히 길을 찾는 데 어려움이 있습니다. 구체적으로, 노인 등 노약자의 경우 여러 편의시설이 비치되어 있으나, 보호자 없이는 여전히 이용하기 쉽지 않습니다.

먼저, 출입자의 얼굴 인식 또는 OTP 기반의 기술을 활용해서 입장을 도와줍니다. 이는 정확한 영상처리를 확인하고 이를 DB에 저장해서 날짜와 시간을 표기됩니다.

이후에 개인별 카드를 통해 원하는 곳을 지정하거나 음성인식에 맞춰 LED로 위치까지 길을 안내해줍니다. 또는 로봇에 탑승하여 목적지까지 자동으로 이송할 수 있습니다. 각각의 위치에는 RFID 리더기에 고유 번호나 이름을 지정하여 구분이 되어있고, 휴대용 GPS를 해당 등록키에 내장된 IMU 모듈을 이용하여 현재 위치 및 방향에 따라 벽면에 배치된 LED를 통해 쉽게 목적지까지 찾아갈 수 있도록 제공합니다.

목적지에 도착한 후에는 방에 들어가기 전에 카드를 인식 하면 그에 맞춰 문이 자동으로 열리고 이용자가 원하는 업무를 진행할 수 있습니다.

## 2. 작품 설명

### 2-1 주요 동작 및 특징

#### 1) 공공시설 입장 : 얼굴 인식 및 OTP 활용

NVIDIA Jetson Nano를 활용하여 실시간으로 사람의 얼굴을 인식하고, 사람인 것을 확인되면 출입구가 열립니다. 사람이 아닌 생물 또는 물건이면 입장이 불가합니다. 또한, 직원 또는 회원이 등록된 경우에는 OTP를 통해 임시 비밀번호로 입장할 수 있습니다.

#### 2) 로봇을 통한 길안내 및 이송

입장한 후에 지급된 카드를 지급합니다. 이후 로봇 실행되는 리더기에 카드를 인식하고, 이용자를 원하는 목적지까지 이송하거나 길안내를 도와준다. 전혀 길을 모르더라도 로봇을 따라가면 안전하게 도착이 가능하게 됩니다.

도착한 후에는 개인별 지급된 카드를 목적지 입구에 인식하거나 또는 10초 뒤에 로봇 주차장으로 자동 복귀하게 됩니다.

#### 3) LED를 통한 길안내

LED가 나오도록 지정된 리더기에 카드를 인식하면 카드로 리더기를 인식하거나, 앱을 이용해 목적지를 설정하면 위치까지 LED 스트립으로 길을 표시해줍니다. 각 도착지점마다 LED 색상을 상이하게 하여 구분이 가능합니다.

#### 4) 음성 인식을 통한 동작






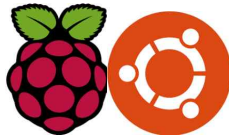

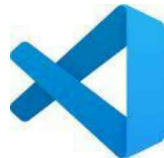
음성 인식을 통해 입장 방문자가 사람인지, 동물인지 또는 같이 동행했는지 여부를 확인할 수 있습니다.

#### 5) 휴대용 기기를 통한 방향 안내



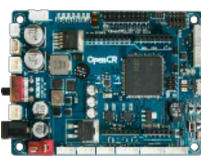





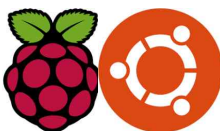


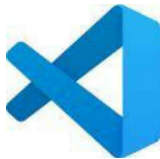
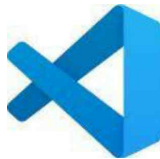
위치 안내해주는 휴대용 기기를 들고 다니면서 사용자가 올바른 방향으로 가는지 확인해줍니다. 따라서 사용자가 길을 잃은 상황이 발생하면, 주변 카드 리더기에 카드를 인식하거나, 직원이 문제를 확인하고 빠르게 도와주러 올 수 있습니다.

## 2-2 전체 시스템 구성

### (1) 전체 개발 환경

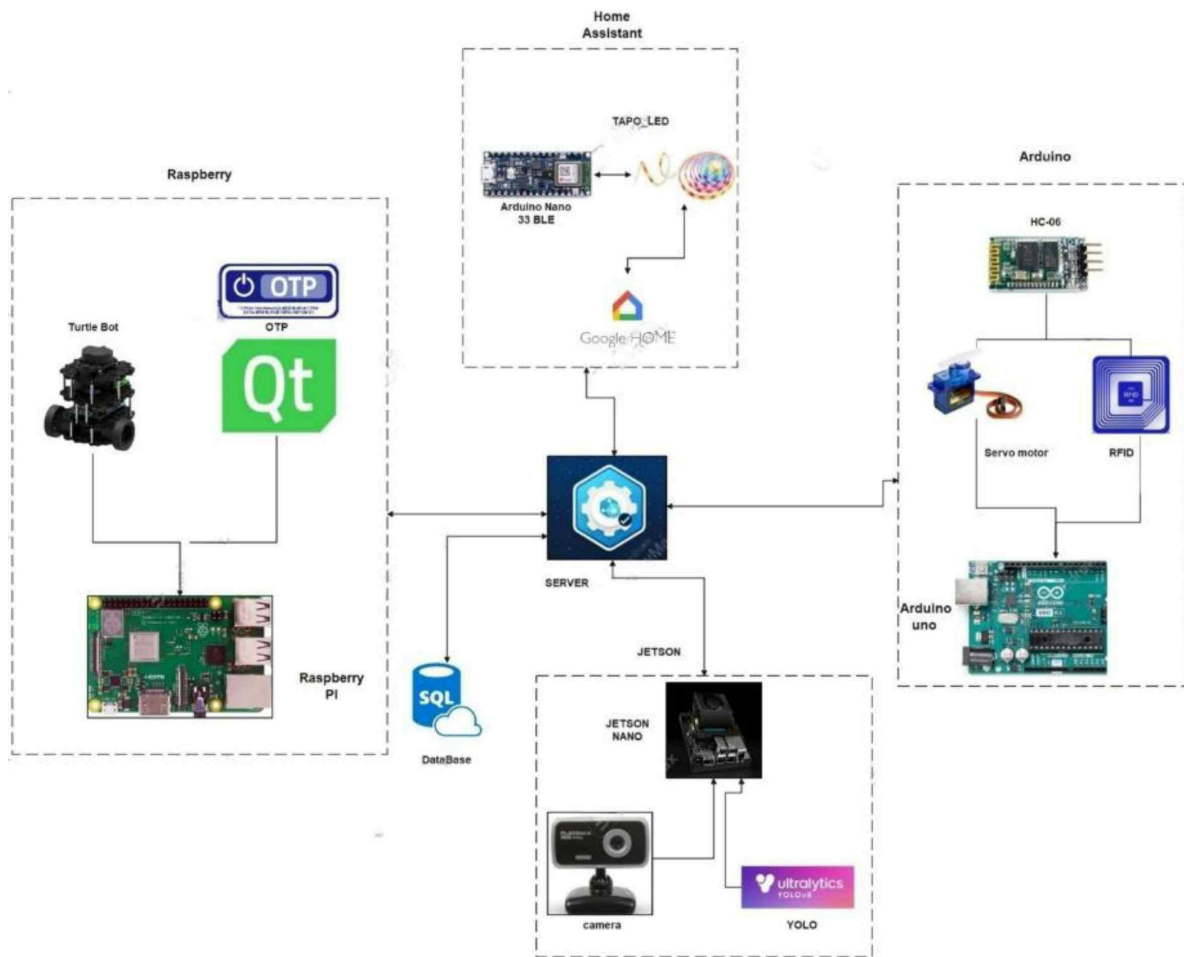
	Main Server	통신 Server
기판	 Seeed Odyssey x86j4105	 Raspberry Pi 4 Model B
언어	 C언어, C++	 C언어
OS	 Ubuntu 20.04	 라즈베리파이 Ubuntu 20.04
Library	Sys/types.h, sys/socket.h fcntl.h, pthread.h, sys/time.h, errno.h	Sys/types.h, sys/socket.h, fcntl.h, pthread.h, sys/time.h, errno.h
시스템	 VS code, DB(HeidiSQL)	 VS code

	Device	얼굴 인식	Turtlebot3
--	--------	-------	------------

기판			
	Arduino Uno Arduino Nano	Jetson Nano	OpenCR, Raspberry Pi 4 Model B
언어			
	C++	Python	Python, C++
OS			
	Windows 11	JetPack ~~~	라즈베리파이 Ubuntu 20.04
Library	SoftwareSerial.h, MFRC522.h, SPI.h, Servo.h, Arduino_BMI270_BMM 150.h		pthread.h, ros/ros.h std_srvs/Setbool.h rosGoalClient/bot3gpio. h
		cv2, socket, gTTS, YOLO	
시스템			
	Arduino IDE	VS code	VS code

(2) 전체 시스템 구성도

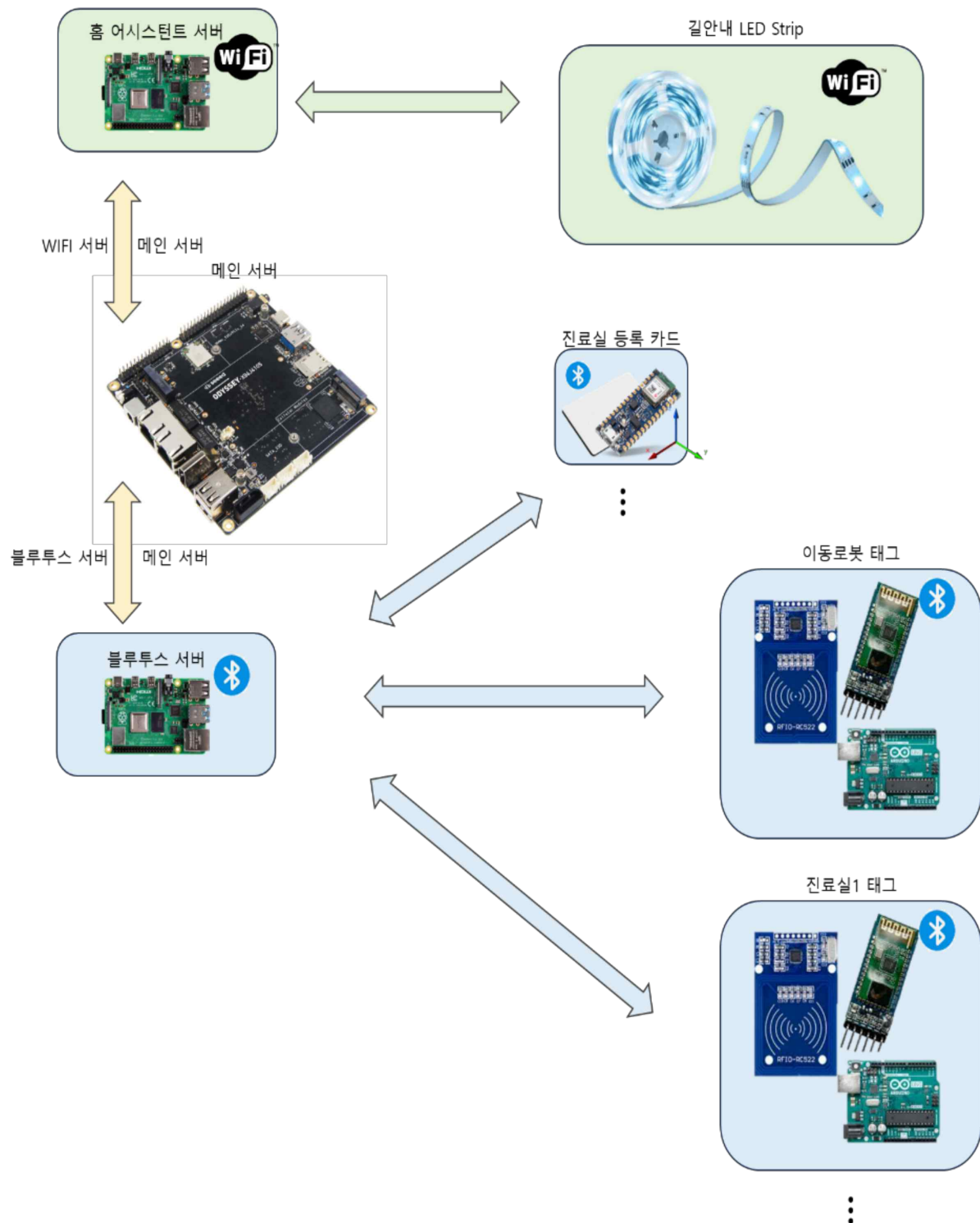
{ }



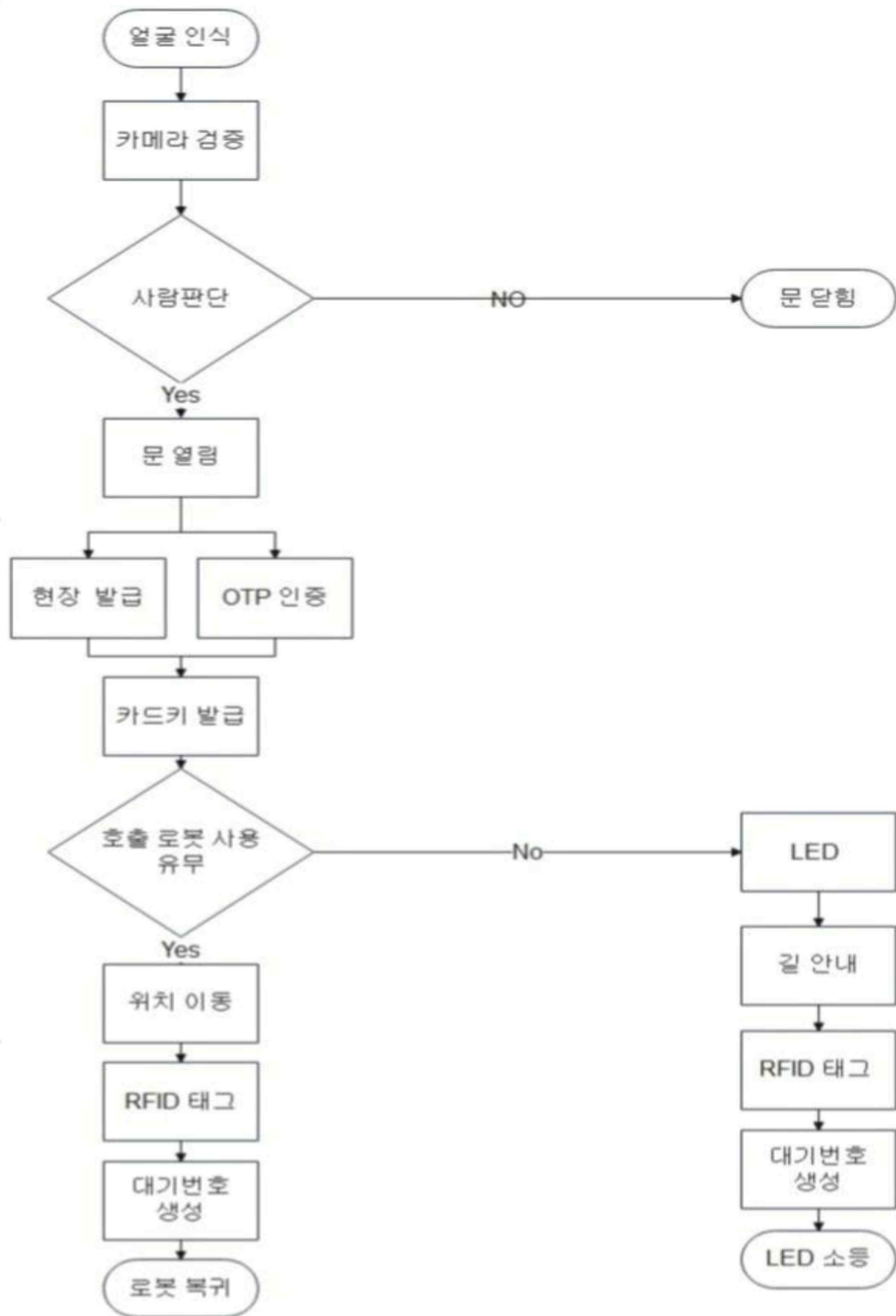
- Arduino Nano 33 BLE가 BLE 비콘 역할을 하며, Home Assistant와 Google Home을 통해 Tapo 스마트 LED 조명을 제어합니다.
- Arduino Uno는 RFID 리더기와 서보모터를 통해 출입 제어 기능을 수행하며, HC-06 블루투스 모듈로 서버와 통신합니다.
- Raspberry Pi는 Turtle Bot, Qt UI, OTP 인증 등을 통해 로봇 제어 및 사용자 인터페이스 기능 수행합니다.
- Jetson Nano는 YOLO 기반 객체 인식을 통해 영상 처리 및 인공지능 기능을 수행합니다.
- 모든 장치는 중앙 서버와 연결되며, 서버는 데이터베이스(SQL)와 연동되어 전체 데이터를 관리합니다.

### (3) 서버 통신 방식





(4) 시스템 Flowchart



## 2-3 개발 환경

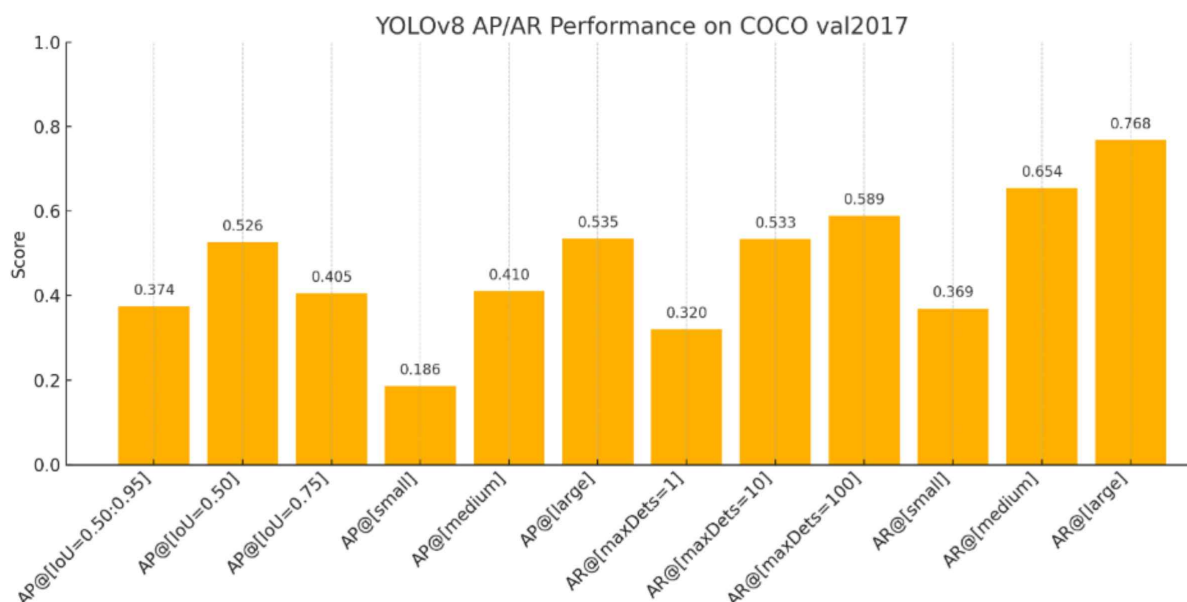
## (1) Jetson Nano

### - Jetson Nano

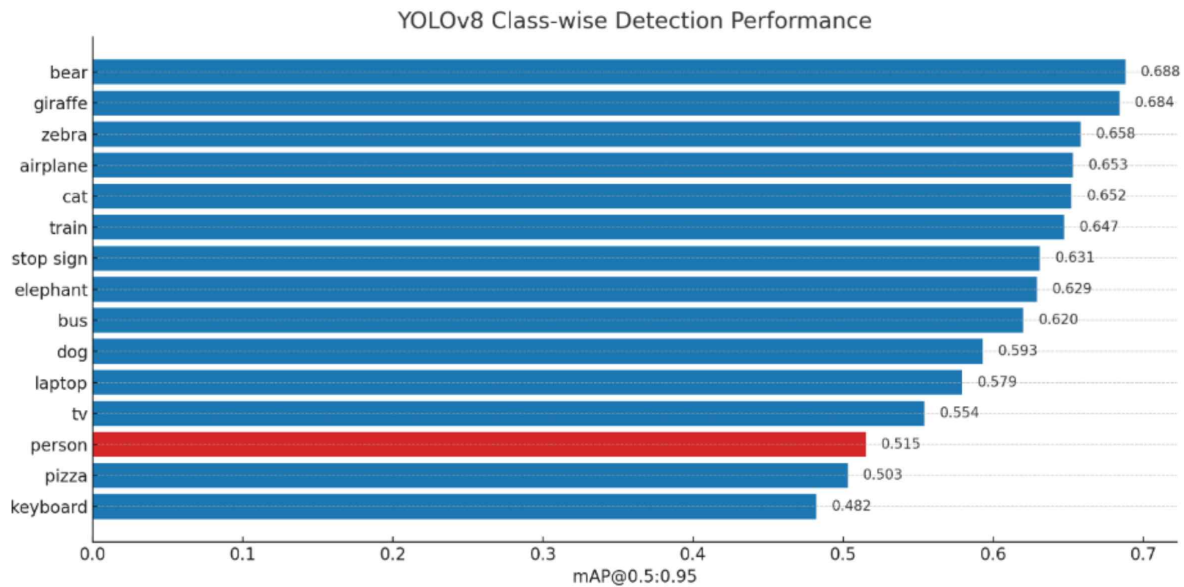
Jetson Nano는 NVIDIA에서 개발한 소형 AI 컴퓨팅 보드로 Edge AI와 임베디드 시스템 개발에 적합합니다. 또한 저전력(5~10W) 환경에서 동작이 가능하고, 다양한 GPIO핀을 활용할 수 있습니다. 따라서 출입 시스템에 필요한 실시간 객체 탐지를 수행하고, TTS를 활용하여 스피커를 통해 음성메시지를 출력할 수 있기에 Jetson Nano를 사용하였습니다.

### - YOLO(You Only Look Once)

YOLO은 Ultralytics 에서 개발한 실시간 객체 탐지에 특화된 인공 지능 모델입니다. YOLO는 말 그대로 한번에 이미지를 보고 객체가 어디있는지 판단하는 데, 기존의 R-CNN계열은 Region Proposal 이후 Classification 두단계를 거쳐 정확한 성능이지만 속도가 느립니다. 하지만 YOLO은 앵커 프리(Anchor-Free) 구조로 정확도가 R-CNN보단 떨어지지만 속도가 빠른 장점이 있습니다. 우리가 사용할 영상처리는 순간의 이미지를 빠르게 판단하여 결과를 도출해야 하기 때문에, YOLO 모델을 선택하였습니다. 그 중 2023년에 출시된 YOLOv8의 객체 탐지 모델을 채택하였고, COCO val2017 데이터셋을 활용하여 모델을 학습시켰습니다. 또한, YOLOv8의 큰 특징은 저 메모리와 적은 연산의 사용으로 객체를 탐지할 때 적합한 모델입니다. <그래프1> <그래프2>



<그래프1. YOLOv8 AP/AR 성능>



<그래프2. YOLOv8 객체별 탐지 성능>

객체 탐색 모델을 평가할때 평균 정밀도(mean Average Precision, mAP) 와 IoU(Intersection over Union)을 기준으로 성능을 평가 하는데 YOLOv8의 mAP@0.5:0.95 Score는 0.374로 낮은 정밀도를 보이는데, 대신 클래스별로 '사람'과 큰 객체의 Score는 각각 0.515, 0.535로 실시간 감지에 활용하기에 충분하고, 동물 클래스의 정밀도는 상당히 높은 성능을 보입니다.

#### - TTS(Text-to-Speech)

본 시스템에서 객체를 탐지하고 결과를 사용자에게 알려주기 위해 TTS 시스템을 적용 하였습니다. TTS는 텍스트 분석 및 음성 합성을 진행하게 되는데, TTS 모델은 gTTS(Google Text-to-Speech) API를 활용한 파이썬 라이브러리를 사용하였습니다. gTTS는 클라우드 기반으로 모델을 로컬에서 학습할 필요가 없고 자연스러운 다국어 음성 생성 기능을 사용할 수 있습니다. 또한, 라이브러리 설치가 간단하고, 동작 구현에 큰 문제가 없어 많은 시스템에 사용됩니다.

#### (3) MFRC522

무선인식(Radio Frequency Identification)이라고 하며, 반도체 칩이 내장된 인식(Tag), 라벨(Label), 카드(Card) 등의 저장된 데이터를 무선 주파수를 이용하여 비접촉으로 읽어내는 인식 시스템입니다.

기존의 QR 코드나 이미지 기반 인식 방식은 조도, 사용자 위치 등에 따라 오작동 가능성이 존재하나, RFID는 다양한 환경에서도 빠르고 안정적으로 고유 정보를 인식할 수 있는 장점이 있습니다. 또한, 저전력, 저비용 기반으로 임베디드 환경에서 안정적인 성능이 가능하며, 다중 리더기 구성 가능하여 구역별 인식 처리에 용이합니다. 마지막으로, 고유 UID 기반의 사용자 구분이 가능하기 때문에 MFRC522를 채택했습니다.

사용자의 동선 인식 및 로봇 호출 트리거의 역할로 활용했습니다. MFRC522 모듈을 사용하여 저전력/저비용으로 시스템을 구현하였으며, 아두이노와 연동하여 Bluetooth 통신으로 서버와 실시간 데이터를 주고받는 구조로 구현했습니다.

### (3) ARDUINO NANO 3.3 BLE Rev2

ARDUINO NANO 3.3 BLE REV2 보드는 ARDUINO사에서 개발한 소형 마이크로 컨트롤러 보드로 웨어러블 장치나 실시간 모션 추적이 필요한 장치를 개발하는데 적합한 보드입니다.

이 기기를 프로젝트에 채택하게 된 몇 가지 특징이 있습니다. 첫번째로 Arm Cortex-M4 CPU를 탑재하여 저전력 환경에서도 안정적인 통신이 가능합니다. 두번째는 Bluetooth LE로 통신이 가능한 Nina B306모듈을 탑재하고 있어 무선 연결을 위해 외부 블루투스 모듈을 사용할 필요가 없고, 사용자 위치 정보를 실시간으로 전송할 수 있습니다. 그리하여 가볍게 사용자가 들고 다닐 수 있습니다. 마지막으로 위치, 방향 및 가속도의 정확한 측정을 제공하는 9축 관성 측정 장치가 장착되어 있습니다. 이를 통해, 이동형 사용자 트래킹과 LED 유도 시스템을 같이 구현했습니다.

### (4) ARDUINO UNO Rev3

ARDUINO UNO Rev3 보드는 ARDUINO사에서 개발한 마이크로 컨트롤러 보드입니다. 마이크로 컨트롤러는 ATMEL사의 ATmega328P를 탑재하고 있으며 오픈소스로 임베디드 시스템을 쉽게 개발할 수 있다는 특징이 있습니다.

이를 통해 RFID 인식, TurtleBot3 호출 신호 송신, 서보 모터 제어를 통한 문 개폐 동작에 사용되었습니다. 이처럼 BLE, 로봇 등 여러 시스템 간 하드웨어 인터페이스를 제어하며, 실제 기기 작동을 담당하는 핵심 컨트롤러로 활용되었습니다.

### (5) Qt

Qt는 컴퓨터 프로그래밍에서 C++ 기반의 GUI 프로그램 개발에 널리 쓰이는 크로스 플랫폼 프레임워크이다. Widget을 드래그 & 드롭 형식으로 추가할 수 있으며, 타 시스템

과의 연동이 용이한 구조로 되어있기 때문에 간결하고 직관적으로 사용자에게 제공함으로써 보안성과 접근성을 동시에 확보할 수 있습니다.

본 프로젝트에서는 OTP(One-Time Password) 입력 시스템 구현에 활용됩니다. 이를 통해 입력된 OTP는 이후 RFID 인증, 로봇 호출 등의 절차를 진행할 수 있습니다.

#### (6) Raspberry Pi 4B

라즈베리 파이 보드는 영국의 라즈베리 파이 재단에서 출시한 ARM 기반의 싱글 보드 컴퓨터로, 크기는 작지만 데스크탑 수준의 성능을 제공하여 다양한 임베디드 시스템, IoT, GUI 기반 프로젝트에 널리 사용됩니다. Wi-fi 기능이 내장돼 있고, LCD도 연결할 수 있어 프로젝트에서 OTP 번호를 입력 받는 장치로 사용했습니다.

#### (7) TAPO - L930-5

- TP-LINK의 스마트 홈 브랜드인 TAPO에서 출시한 스마트 LED 스트립 라이트입니다. 이는 다양한 색상 표현과 개별 LED 제어 기능을 제공하며, Home Assistant와 연동하여 실시간 제어가 가능합니다. 라즈베리 파이에 Home Assistant 와 Arduino Nano 33 BLE를 이용하여 주변에 있는 라즈베리 파이에서 감지 MQTT 애드온을 통해 실시간 통신을 합니다. 이 특성을 통해 프로젝트에서 사용자가 이동할 목적지 방향을 시각적으로 안내하기 위한 인터랙티브 표시 장치로 활용했습니다. RFID의 값을 인식하면 소등하게 설정하였습니다.

#### (8) DataBase

해당 데이터베이스는 의료기관이나 대형 시설에서 환자/민원인의 이동을 돕고 출입을 관리하는 통합 시스템으로 설계되었습니다. 사용자 인증, 보조이동수단 대여, 출입 카드 관리가 유기적으로 연결되어 있어 효율적인 시설 관리가 가능합니다.

### 3. 단계별 제작 과정


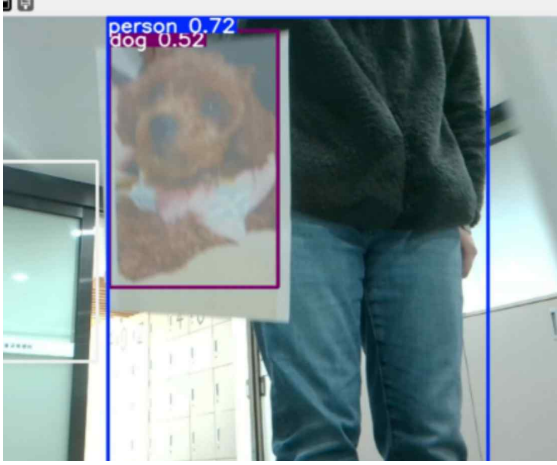
#### (1) 서버

각 클라이언트에서 요청을 받으며 기능을 실행하거나 받은 데이터를 다른 클라이언트에 전달한다. 변경된 내용을 데이터베이스에 반영함

#### (2) Jetson

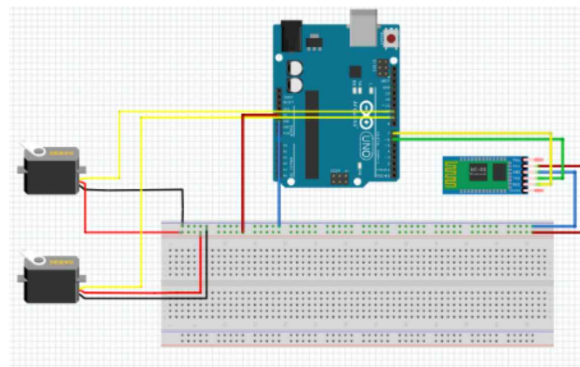
##### - Jetson 동작

YOLO의 yolov8n.pt 데이터셋을 모델로 지정한 후, 서버와의 통신을 위해 서버 IP, 포트, 클라이언트 ID, 데이터를 보내기 위한 SEND\_ID를 설정합니다. 카메라에 사람이 2초간 감지되면 SEND\_ID로 "senser@human"을 전송하고, 스피커를 통해 "사람이 감지되었습니다. 문이 열립니다."를 출력합니다. 또한 출입문에 연결된 아두이노에 문열림 메시지를 보내어 문이 10초간 열렸다가 문이 닫히게 됩니다. 만약 동물이 함께 있다는 것이 감지되면 동물의 종류와 함께 SEND\_ID로 "senser@dog" 와 같이 메시지를 전송하고, 스피커를 통해 "동물이 감지되었습니다. 관리실에 문의해주세요"를 출력하게 되며 출입문이 열리지 않습니다. <사진 1>, <사진 2>

	
<사진1. YOLO로 사람 인식>	<사진2. YOLO로 동물과 사람 인식>

### (3) Servo 모터를 통한 문 개폐 시스템

#### - 회로도



<사진 3 : Servo 모터 회로도>

#### - 동작 설명

사람만 출입이 되면(사진1 참고) 방문자가 들어올 수 있게 문을 개방합니다. 문을 개방하는 것은 아두이노 보드에 서보 모터 두개를 연결했으며, Jetson과 블루투스, 아두이노를 통해 이루어진다. 사람 인식이 끝난 후에 "DOOR@OPEN" 명령을 서버에 보내면 그에 맞춰 개방되고 5초 뒤 자동으로 "DOOR@CLOSE" 명령을 통해 문이 닫히게 됩니다.

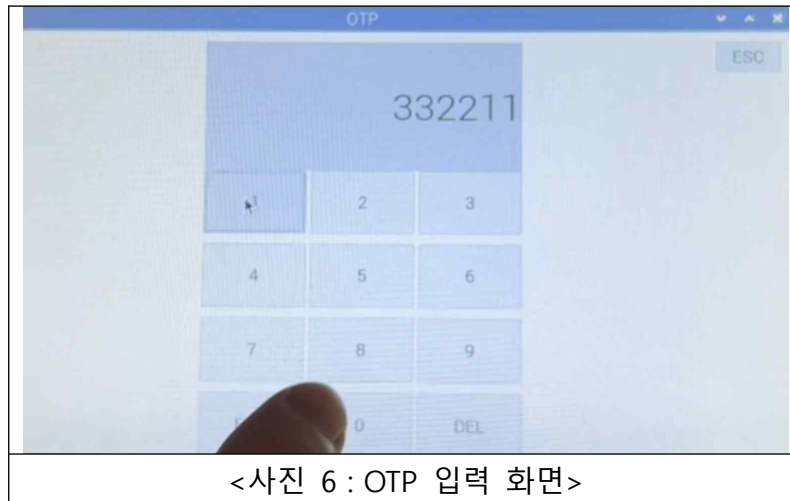
<pre>msg : [PJ_AP-&gt;PJ_DOOR] DOOR@OPEN msg : [PJ_AP-&gt;PJ_DOOR] DOOR@CLOSE</pre>	<pre>SRV:[PJ_AP]DOOR@OPEN SRV:[PJ_AP]DOOR@CLOSE ARD:Door closed to 0 degrees.</pre>
<사진 4 : 서버의 문열림과 닫힘 신호>	<사진 5 : 출입문 블루투스 클라이언트>

### (4) OTP 입력 디스플레이 구현

입장 후 방문자는 길안내 서비스 사전 예약을 했다면 OTP를 통해 본인 인증을 진행합니다. 방문자는 본인 인증 후 TurtleBot3를 통한 이송 서비스 또는 GPS(LED와 아두이노 나노)를 선택해 길안내를 받을 수 있습니다.

누구나 글씨를 알아보기 쉽고 터치로 즉시 입력이 가능하도록 Qt를 사용했습니다. 빌드된 실행파일은 방문자가 OTP를 입력하고 ENT 버튼을 누르면 OTP번호가 서버로 전송돼 검증합니다. 이후 OTP 정보가 일치하면 방문자는 카드를 지정 받아 사용할 수 있습니다.





<사진 6 : OTP 입력 화면>

- DB 부분

<pre>[PJ_AP]REGISTER@332211 SELECT userID FROM UserTable where OTP="332211"; [PJ_AP]REGISTER@332211@1</pre>	<pre>[PJ_AP]USE_CARD@1 SELECT CardNumber FROM Users where isBusy=0 [PJ_AP]USE_CARD@1@7973A698 [PJ_AP]USE_GUIDE@1</pre>
<사진 7 : 서버 - OTP 등록>	<사진 8 : 카드번호 지정>

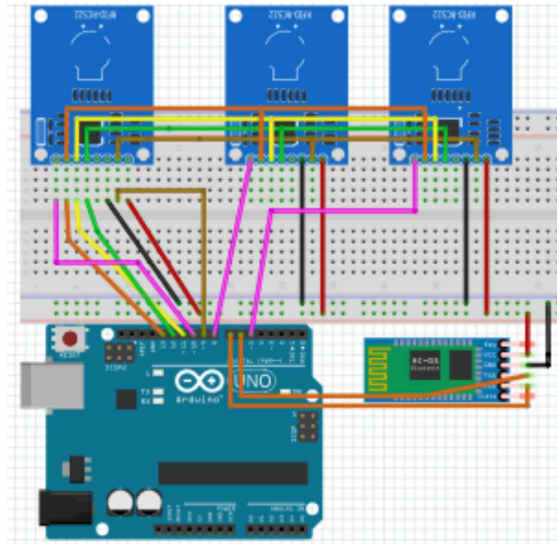
UserID	CardNumber	GuideNumber	Room	Order	OTP	Authenticated	AccessLevel	CurrentSection
1	7973A698	1	2	-1	332211	1	3	

<사진 9 : DB 등록된 현황>

사진과 같이 방문자가 사전에 목적지를 설정하고, 그에 맞는 "332211" OTP 번호를 입력하면 서버와 DB에 OTP 번호가 전송되며, ROOM2로 가게 된다는 것을 확인할 수 있습니다. <사진 7>, <사진 8>, <사진 9>

### (5) RFID를 통한 인식

#### - 회로도



< 사진 10 : MFRC522 회로도 >

#### - 구동 방법

SPI 통신을 사용하기 때문에 여러 개 모듈을 사용해서 SS를 아두이노에서 핀을 지정하여 연결함으로써, TCP 소켓 기반으로 작성했으며, 여러 클라이언트들과 동시에 연결한다. 방문자가 도착 후 위치별 지정된 RFID Reader 번호와 UID 데이터를 SQL 클라이언트로 안정적으로 전송한다. 블루투스를 통해 들어온 RFID 정보를 서버에서 포워딩 받아, 실시간으로 SQL 데이터베이스에 저장된다. <사진11>, <사진12>, <사진13>

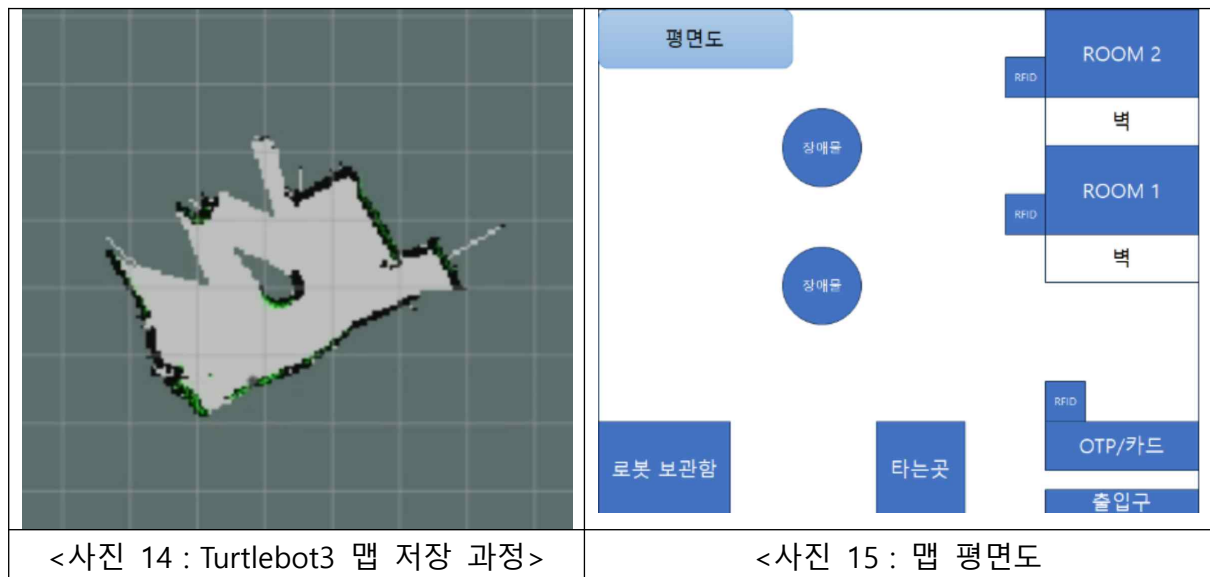


### (3) Turtlebot3(ROS)

Turtlebot3는 원격 PC에 Ubuntu 20.04버전의 Linux 환경을 구축합니다. 이후 OpenCR에

ROS 환경을 설치하고 .bashrc 환경설정에서 ROS\_MASTER\_URI와 ROS\_HOSTNAME을 원격 PC환경에 맞게 수정합니다. 이후 원격 PC에 "roscore"실행, Turtlebot3에 "roslaunch turtlebot3\_bringup turtlebot3\_robot.launch"를 실행하여 Turtlebot3의 센서, 모터 노드등을 실행하게 됩니다.

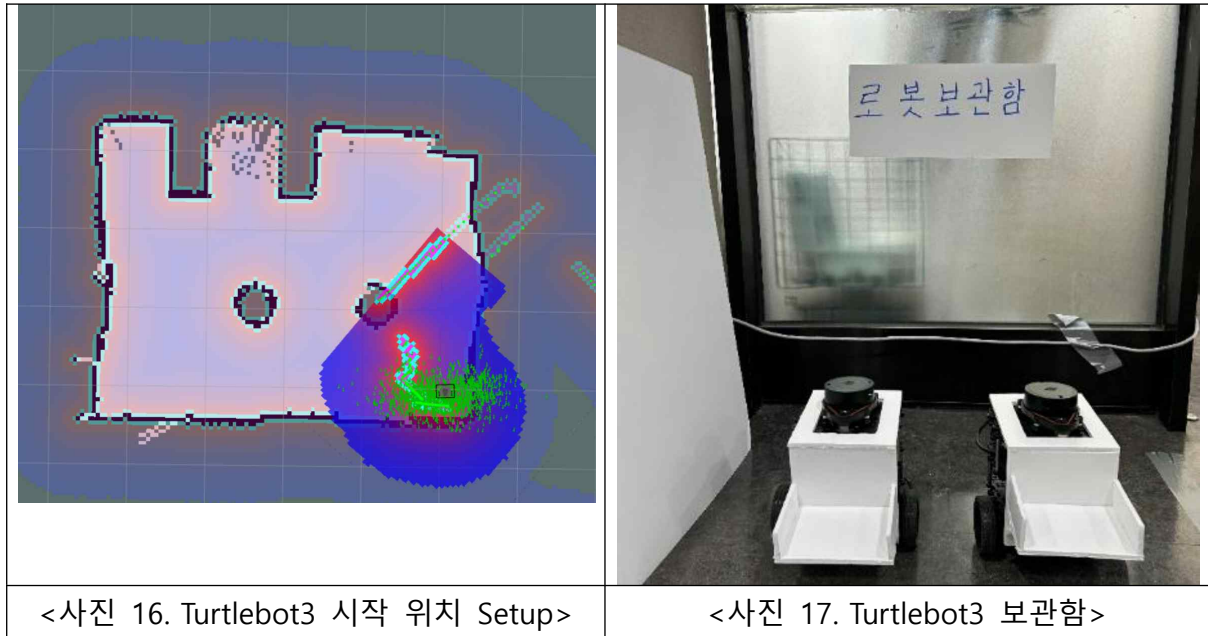
Turtlebot3에 맵을 저장하기 위해 원격 PC에 "roslaunch turtlebot3\_teleop turtlebot3\_teleop\_key.launch"를 통해 원격으로 Turtlebot3를 제어한 후, 시작지점에 Turtlebot3를 두고, "roslaunch turtlebot3\_slam turtlebot3\_slam.launch"을 실행하여 맵을 저장할 준비를 하게 됩니다.



맵 전체를 Turtlebot3를 통해 저장했다면 "roslaunch turtlebot3\_navigation turtlebot3\_navigation.launch map\_file:=\$HOME/map/PJ\_map.yaml" 을 실행하여 저장한 맵에서 Turtlebot3이 인식 되는 지 확인 후에 "rostopic echo /amcl\_pose" 명령어를 통해 Turtlebot3의 현재 위치와 좌표를 파악한 후, 원하는 곳의 좌표를 저장합니다.

이후 클라이언트를 통해 서버와 연결 후, 사용자가 RFID카드를 인식하게 되면 Turtlebot3이 사용자 앞으로 대기 하게되고, 사용자가 원하는 곳까지 안내하게 됩니다.

<사진14> , <사진 15>, <사진 16>, <사진 17>



#### - 시연 동작 구성

터틀봇을 통한 이송이 필요한 사용자의 경우, 카운터에서 OTP 인증 시 해당 카드키를 제공받습니다. 이후 로봇 보관함에 있는 Turtlebot3이 해당 위치로 하며, 이동 후 3초간 대기하며, 그 시간에 사용자가 탈 수 있습니다. ROOM2에 위치한 후 5초후 다시 로봇보관함으로 돌아갑니다. <사진 18>, <사진 19>, <사진 20>



#### - DB

##### ① 방문자가 Turtlebot3 호출 RFID 인식 전

- 인식 전에는 USERID와 isBusy 부분의 값이 101, 0 으로 되어있으며, 이동로봇 사용 현황도 두번째 줄이 101, 0으로 되어있습니다. <사진21>, <사진22>

카드 사용 현황		
Users: 5 행 (총) (정확한) >> 다음 << 모두 보기 < 정렬		
CardNumber	UserID	isBusy
-1	(NULL)	1
098CA298	2	1
137337FC	1	1
391FA298	4	1
7973A698	101	0

이동로봇 사용 현황		
Facilities: 4 행 (총) (정확한) >> 다음 << 모두 보기 < 정렬		
GuideNumber	UserID	isBusy
-1	1	1
1	101	0
2	101	1
3	101	1



<사진 22 : 초기카드 사용 현황>

사용자 시설 이용 현황								
Facilities: 6 행 (총) (정확한) >> 다음 << 모두 보기 < 정렬 < 열 (9/9)								
UserID	CardNumber	GuideNumber	Room	Order	OTP	Authenticated	AccessLevel	
1	-1	-1	2	-1	332211	0	3	

<사진 23 : 초기 사용자 시설 이용 현황>

- 인식 후에는 USER\_CARD의 값이 서버와 DB로 전송되어 로봇 현황과 카드 사용 현황이 사진과 같이 값이 1과 1로 변경됩니다. <사진 24>, <사진 25>, <사진 26>, <사진 27>

msg : [PJ\_SQL->PJ\_AP] REGISTER@332211@1  
msg : [PJ\_AP->PJ\_SQL] USE\_CARD@1  
msg : [PJ\_SQL->PJ\_AP] USE\_CARD@1@7973A698  
msg : [PJ\_AP->PJ\_RFID@] READ@2@7973A698  
msg : [PJ\_AP->PJ\_BOT1] READ@2@  
msg : [PJ\_AP->PJ\_SQL] USE\_GUIDE@1  
msg : [PJ\_SQL->PJ\_AP] USE\_GUIDE@1@1

GuideNumber	 UserID	 isBusy
-1	1	1
1	1	1
2	101	1
3	101	1

< 사진 24 : 서버 - Turtlebot3 호출 RFID 인식>

<사진 25 : Heidi - 로봇 사용현황>

```
[PJ_AP]REGISTER@332211@1
[PJ_AP]USE_CARD@1
SELECT CardNumber FROM Users where isBusy=0
[PJ_AP]USE_CARD@1@7973A698
[PJ_AP]USE_GUIDE@1
SELECT GuideNumber FROM Facilities where is
Busy=0;
[PJ_AP]USE_GUIDE@1@1
```

ers: 5 행 (총) (정확한) >>

CardNumber	UserID	isBusy
-1	(NULL)	1
098CA298	2	1
137337FC	1	1
391FA298	4	1
7973A698	1	1

<사진 26 : DB - Turtlebot3 호출 RFID 인식>

<사진 27: Heidi – 카드 사용 현황>

- 이송 후 Turtlebot3이 복귀하면 FREE\_GUIDE@1 신호를 보내면서 로봇이 다른 방문자를 이송할 수 있는 상태로 변경됩니다. <사진28>, <사진29>, <사진30>, <사진31>

<pre>msg : [PJ_AP-&gt;PJ_SQL] FREE_GUIDE@1 msg : [PJ_SQL-&gt;PJ_AP] FREE_GUIDE@1 msg : [PJ_AP-&gt;PJ_SQL] FREE_CARD@7973A698 msg : [PJ_SQL-&gt;PJ_AP] FREE_CARD@7973A698</pre>	<table><tr><th>UserID</th><th>CardNumber</th><th>GuideNumber</th><th>Room</th><th>Order</th></tr><tr><td>1</td><td>-1</td><td>-1</td><td>2</td><td>1</td></tr><tr><td>2</td><td>098CA298</td><td>-1</td><td>1</td><td>-1</td></tr><tr><td>3</td><td>137337FC</td><td>-1</td><td>2</td><td>-1</td></tr><tr><td>4</td><td>391FA298</td><td>-1</td><td>1</td><td>-1</td></tr><tr><td>101</td><td>-1</td><td>-1</td><td>-1</td><td>-1</td></tr><tr><td>1,001</td><td>-1</td><td>-1</td><td>-1</td><td>-1</td></tr></table>	UserID	CardNumber	GuideNumber	Room	Order	1	-1	-1	2	1	2	098CA298	-1	1	-1	3	137337FC	-1	2	-1	4	391FA298	-1	1	-1	101	-1	-1	-1	-1	1,001	-1	-1	-1	-1
UserID	CardNumber	GuideNumber	Room	Order																																
1	-1	-1	2	1																																
2	098CA298	-1	1	-1																																
3	137337FC	-1	2	-1																																
4	391FA298	-1	1	-1																																
101	-1	-1	-1	-1																																
1,001	-1	-1	-1	-1																																
<사진 28 : 서버 - 로봇 복귀 시>	<사진 29 : Heidi - 사용자 시설 이용현황>																																			
<pre>[PJ_AP]FREE_CARD@7973A698 select userID From Users where CardNumber="7973A698 ";</pre>	<table><tr><th>GuideNumber</th><th>UserID</th><th>isBusy</th></tr><tr><td>-1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>101</td><td>0</td></tr><tr><td>2</td><td>101</td><td>1</td></tr><tr><td>3</td><td>101</td><td>1</td></tr></table>	GuideNumber	UserID	isBusy	-1	1	1	1	101	0	2	101	1	3	101	1																				
GuideNumber	UserID	isBusy																																		
-1	1	1																																		
1	101	0																																		
2	101	1																																		
3	101	1																																		
<사진 30 : DB - 로봇 복귀 시>	<사진 31 : Heidi - 로봇 사용현황>																																			

#### (4) LED 제어



< 사진 32 : LED 연결 방식>

길안내 시스템은 Arduino Nano 33 BLE를 기반으로 한 BLE 비콘과 Raspberry Pi를 활용한 BLE 수신 시스템을 통해 사용자의 실내 위치를 인식하고, Tapo 스마트 LED 스트립 조명을 자동으로 제어하는 스마트 환경을 구축했습니다.

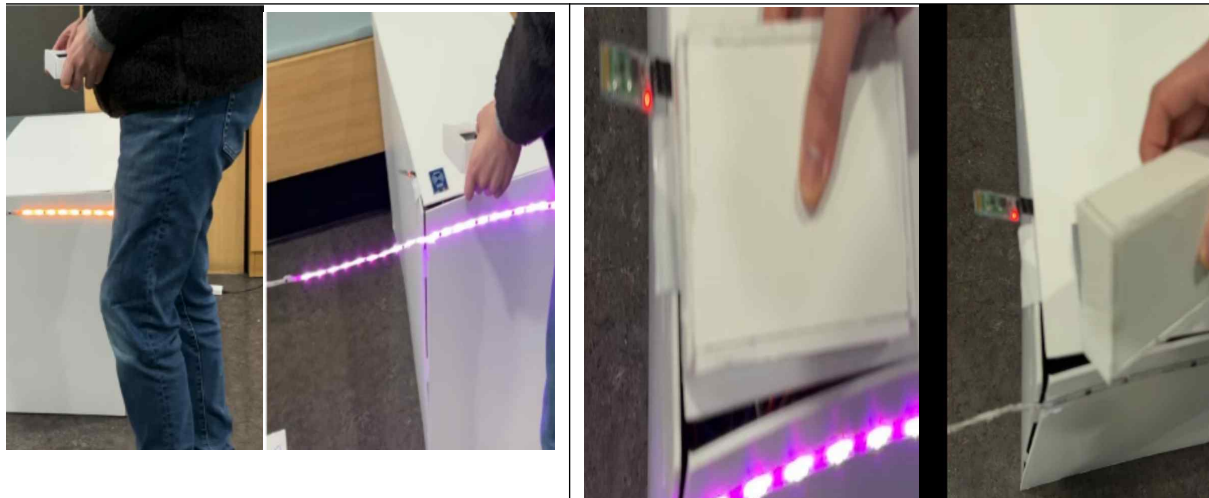
사용자가 휴대하는 Arduino Nano 33 BLE는 BLE 신호를 지속적으로 송출하며, 복도 구간에 배치된 Raspberry Pi는 내장된 Bluetooth 모듈을 통해 해당 신호의 세기(RSSI)를 감지합니다.

#### - 구동 방식

사람이 BLE 신호를 송출하는 등록 수신기를 들고 있으면, 카드가 현재 위치 3번 구간에



있음을 확인한다. 이후 BLE 수신기 2개(양 끝)에서 신호 세기(RSSI) 비율 7:3으로 위치 추정하며, Home Assistant와 서버가 이 위치값을 이용해 동작 제어 가능하다. <사진 33>






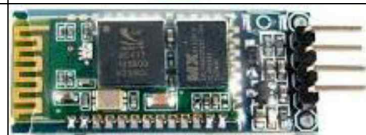








<사진 33 : 길안내를 나타내는 LED>

<사진 34 : RFID 인식 후 소등>

가장 가까운 위치에 설치된 Tapo 스마트 LED 조명을 자동으로 점등하고, 이전 구간의 조명은 순차적으로 소등되도록 구성되어 있습니다. 또한, 사용자가 목적지에 도착한 후 RFID 리더기에 인식을 인식시키면 해당 조명이 자동으로 소등되도록 자동화 설정을 구현했습니다. <사진 34>

#### 4. 사용한 제품 리스트

제품명	이미지	링크
Arduino Nano 33 BLE Rev2		<a href="https://www.devicemart.co.kr/goods/view?no=12229353">https://www.devicemart.co.kr/goods/view?no=12229353</a>
Arduino Uno R3		<a href="https://www.devicemart.co.kr/goods/view?no=34404">https://www.devicemart.co.kr/goods/view?no=34404</a>
Raspberry Pi 4 Model B		<a href="https://www.devicemart.co.kr/goods/view?no=12234534">https://www.devicemart.co.kr/goods/view?no=12234534</a>
NVIDIA Jetson Nano Development Kit-B01		<a href="https://www.devicemart.co.kr/goods/view?no=12513656&amp;srsId=AfmBOorgRc9m_705OliKXlJFKwh4iNL1O1JvM1Ot4tt5cBnG06SsshE_">https://www.devicemart.co.kr/goods/view?no=12513656&amp;srsId=AfmBOorgRc9m_705OliKXlJFKwh4iNL1O1JvM1Ot4tt5cBnG06SsshE_</a>
SG90-HV Digital Servo Motor		<a href="https://www.devicemart.co.kr/goods/view?no=12503476">https://www.devicemart.co.kr/goods/view?no=12503476</a>
HC-06 (DIP) [SZH-EK010]		<a href="https://www.devicemart.co.kr/goods/view?no=1278220">https://www.devicemart.co.kr/goods/view?no=1278220</a>
RFID-RC522		<a href="https://www.devicemart.co.kr/goods/view?no=1279308">https://www.devicemart.co.kr/goods/view?no=1279308</a>
SMPS, AC 90~264V, 350W, 12V 29A		<a href="https://www.devicemart.co.kr/goods/view?no=12710311">https://www.devicemart.co.kr/goods/view?no=12710311</a>
TAPO L930-5		<a href="https://www.devicemart.co.kr/goods/view?no=15075284">https://www.devicemart.co.kr/goods/view?no=15075284</a>
보조배터리 10000mah		<a href="https://www.coupang.com/vp/products/8181394318?itemId=23392419402&amp;vendorItemId=90419850493&amp;sourceType=SDP_MID_CAROUSEL_2&amp;clickEventId=58ffd480-fe4e-11ef-80e1-e760c9d9217f&amp;templateId=3880&amp;isAddedCart=">https://www.coupang.com/vp/products/8181394318?itemId=23392419402&amp;vendorItemId=90419850493&amp;sourceType=SDP_MID_CAROUSEL_2&amp;clickEventId=58ffd480-fe4e-11ef-80e1-e760c9d9217f&amp;templateId=3880&amp;isAddedCart=</a>
PLEOMAX W-210		<a href="https://www.guidecom.co.kr/shop/detail.html?gid=161061">https://www.guidecom.co.kr/shop/detail.html?gid=161061</a>
Turtlebot3		<a href="https://eduino.kr/product/detail.html?product_no=4766&amp;nbsrc=adwords_x&amp;nbkw=&amp;gad_source=4&amp;gclid=Cj0KCQjwy46_BhDOARisAlvmcwO-qFw45P1xvUs2HU2LVNunO-QaOUikv1nDS9rAdHKdu-i6kdr8tWgaAgobEALw_wcB">https://eduino.kr/product/detail.html?product_no=4766&amp;nbsrc=adwords_x&amp;nbkw=&amp;gad_source=4&amp;gclid=Cj0KCQjwy46_BhDOARisAlvmcwO-qFw45P1xvUs2HU2LVNunO-QaOUikv1nDS9rAdHKdu-i6kdr8tWgaAgobEALw_wcB</a>



## 5. 추후 개발 요소

### 1) OTP 인터페이스 개선

현재 적용된 OTP(One-Time Password) 시스템의 사용자 인터페이스를 개선하여, 보다 직관적이고 편리하게 인증할 수 있도록 기능을 보완할 예정입니다.

특히 다양한 환경에서도 원활하게 인증이 가능하도록 반응성과 사용자 경험(UX)을 향상시키는 것이 목표입니다.

### 2) 길안내 장치의 디바이스 추가

사용자의 이동 경로를 실시간으로 추적, 지정된 경로를 이탈할 경우에는 로봇이 음성 안내 기능을 통해 사용자를 다시 정해진 경로로 유도할 수 있도록 안내 장치를 추가할 예정입니다.

또한, 로봇에는 기본적인 음성 안내 언어 기능을 제공하여 한국어(다국어)를 기능 확장이 가능합니다. 추가적으로, 사용자와의 상호작용을 위한 물리적 버튼 기능을 도입하여 아래와 같은 동작이 가능해질 것으로 예상됩니다.

“기본 안내 시작” 버튼

“대기 요청” 버튼

“응급 호출” 버튼

### 3) 데이터 객체화 및 확장 적용

현재 시스템의 안내 정보를 데이터 객체화하여, 향후 다른 관공서, 공공시설, 병원, 안내가 필요한 장소 등에서도 쉽게 적용 가능하도록 모듈화된 적용 방식을 검토 중입니다. 이를 통해 다양한 공간에 맞춤형 길안내 시스템을 구축할 수 있습니다.

### 4) 어플리케이션 연동 기능

해당 길안내 시스템을 모바일 애플리케이션과 연동하여, 사용자가 스마트폰을 통해 실시간 안내를 받을 수 있도록 기능을 확장할 계획입니다. 또한, Google OTP API와의 연동을 통해 다양한 위치에서도 인증 및 접근이 가능하게 하여, 시스템의 범용성과 편의성을 높이하고자 합니다.

## 6. 참고문헌

- 임태훈, 표윤석, 『ROS 2로 시작하는 로봇 프로그래밍』 루비페이퍼, 2021
- 서영진, 『사물인터넷을 위한 리눅스 프로그래밍 with 라즈베리 파이』, 제이펍, 2020
- 황선규, 『OpenCV 4로 배우는 컴퓨터 비전과 머신 러닝』, 길벗, 2019
- <https://emanual.robotis.com/>
- 통계청 「2023년 장래인구특별추계: 시도편」
- 한국지능정보사회진흥원 「2021 디지털정보격차실태조사」
- QT :  
[https://ko.wikipedia.org/wiki/Qt\\_%28%EC%86%8C%ED%94%84%ED%8A%B8%EC%9B%A8%EC%96%B4%29](https://ko.wikipedia.org/wiki/Qt_%28%EC%86%8C%ED%94%84%ED%8A%B8%EC%9B%A8%EC%96%B4%29)
- 아두이노 나노 :  
<https://www.icbanq.com/P015530240srsItid=AfmBOorAUcwmb1BiFJEF2sDJHNfHH3kqdJkQTfBs4KKfkT3mOLrmJZII>
- 소스 코드

첨부파일 참고