

과목명	IoT소프트웨어응용				
평가 내용	2023년도 2학기 기말 프로젝트 최종 보고서				
프로그램명	컨테이너 선적 자동화 시스템				
프로그램 내용	<p>항만 하역장 인명피해가 증가하면서 이러한 자동화된 시스템은 안전을 고려하여 설계됩니다. 여러 센서는 작업 중 안전을 유지하기 위해 컨테이너 및 주변 환경을 탐지하고 적절한 조치를 취합니다. 수동적인 컨테이너 선적 작업에서 발생하는 사고나 인적 오류 가능성을 최소화합니다. 이러한 자동화된 시스템은 사람의 개입을 줄여 안전성을 향상시킵니다.</p>				
주요 기능	<ul style="list-style-type: none"> - 라인트레이서 인식 - QR코드 인식 - 초음파센서 물건감지 - 웹서버 데이터 전송 				
주요 라이브러리	SoftwareSerial, WiFiEsp				
프로젝트 수행 기간	2023.11.2~2023.12.14				
학생 정보	팀장	학번	201904020	이름	윤건용
	팀원	학번	201904026	이름	이호규
		학번	202104035	이름	황수민

1. 배경

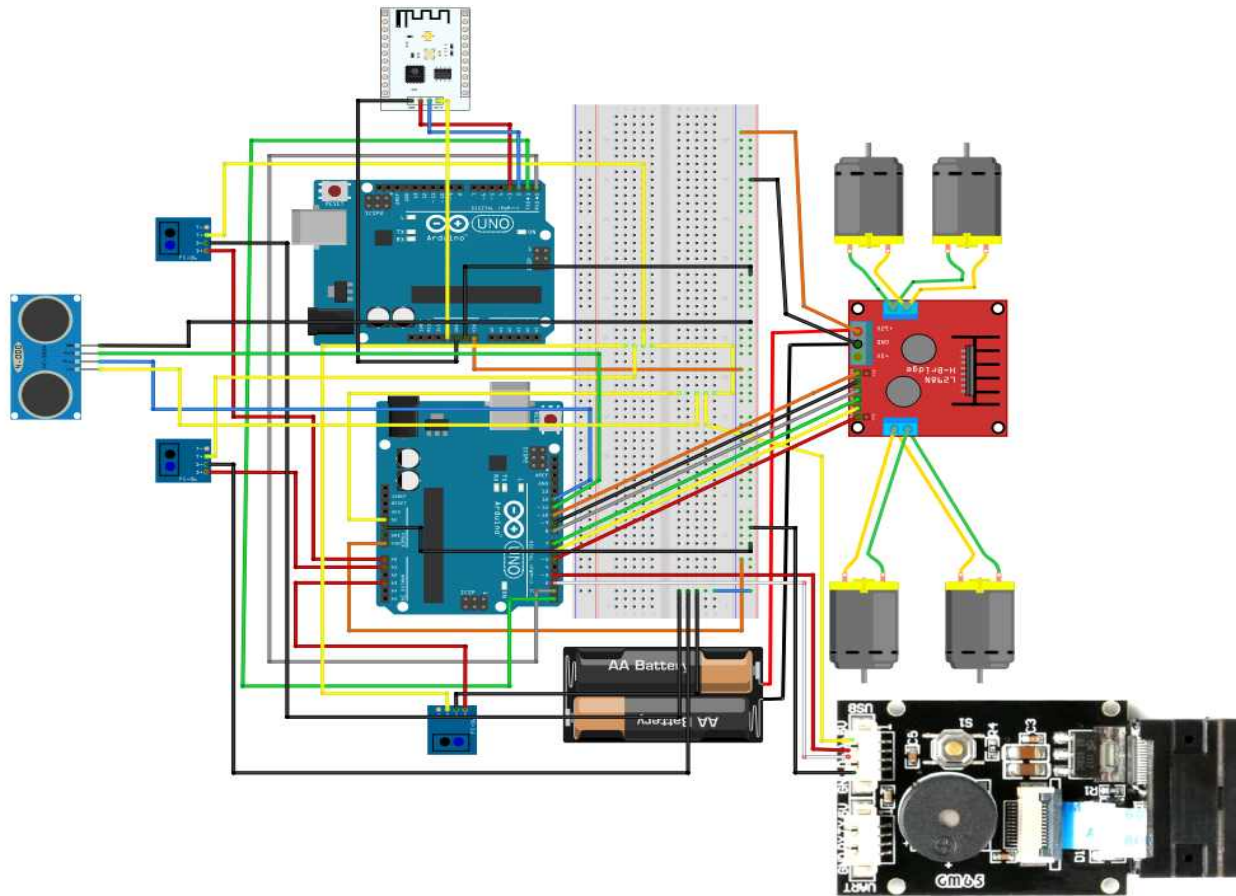
최근 10년간(2013년~2022년) 항만 하역장 내 사고로 2857명의 재해자가 발생하였으며 44명의 사망자가 있는 것으로 확인되었고 지난해 항만 재해자가 351명으로 2017년 대비 59.5% 증가하였습니다. 본 프로그램에서의 자동화 시스템은 컨테이너를 자동으로 선적하고 관리함으로써 작업자들의 안전을 보장하고, 인명피해를 최소화하는 데 중점을 두고 있습니다. 컨테이너 선적 자동화 시스템은 컨테이너를 선적 및 하역하는 작업을 자동화하고

관리함으로써 작업의 편리함과 안전함을 보장하고, 기기의 이동과 컨테이너 식별 등의 부가적인 기능 또한 자동화하여 작업을 효율적이고 편리하게 수행할 수 있도록 지원하도록 하였습니다.

2. 프로그램 기대효과

- 안전성 향상: 자동화된 시스템은 작업자들의 안전을 강화합니다. 작업 중 발생할 수 있는 위험을 줄여 인명피해를 최소화하고, 안전사고 발생 가능성을 감소시킵니다.
- 효율적인 작업 수행: 자동화 시스템은 반복적이고 정확한 작업을 가능하게 합니다. 이를 통해 작업 시간을 단축하고, 선적 및 해운 작업의 효율성을 높일 수 있습니다.
- 자원 최적화: 프로그램을 통해 자동화된 선적 시스템은 자원을 효율적으로 활용할 수 있습니다. 예를 들어, 공간 활용을 최적화하여 선적량을 극대화하거나, 에너지 소비를 줄이는 등의 효과를 기대할 수 있습니다.
- 고객 만족도 향상: 물류 작업의 정확성과 신속성이 증가함에 따라 고객 서비스 품질이 향상됩니다. 정확한 선적 및 배송은 고객들에게 더 신속하고 정확한 물류 서비스를 제공할 수 있습니다.
- 비용 절감: 자동화된 프로세스는 인력 및 시간을 절약할 수 있어 비용을 절감할 수 있습니다. 또한, 더 효율적으로 자원을 활용함으로써 운영 비용을 최소화할 수 있습니다.

3. 회로도



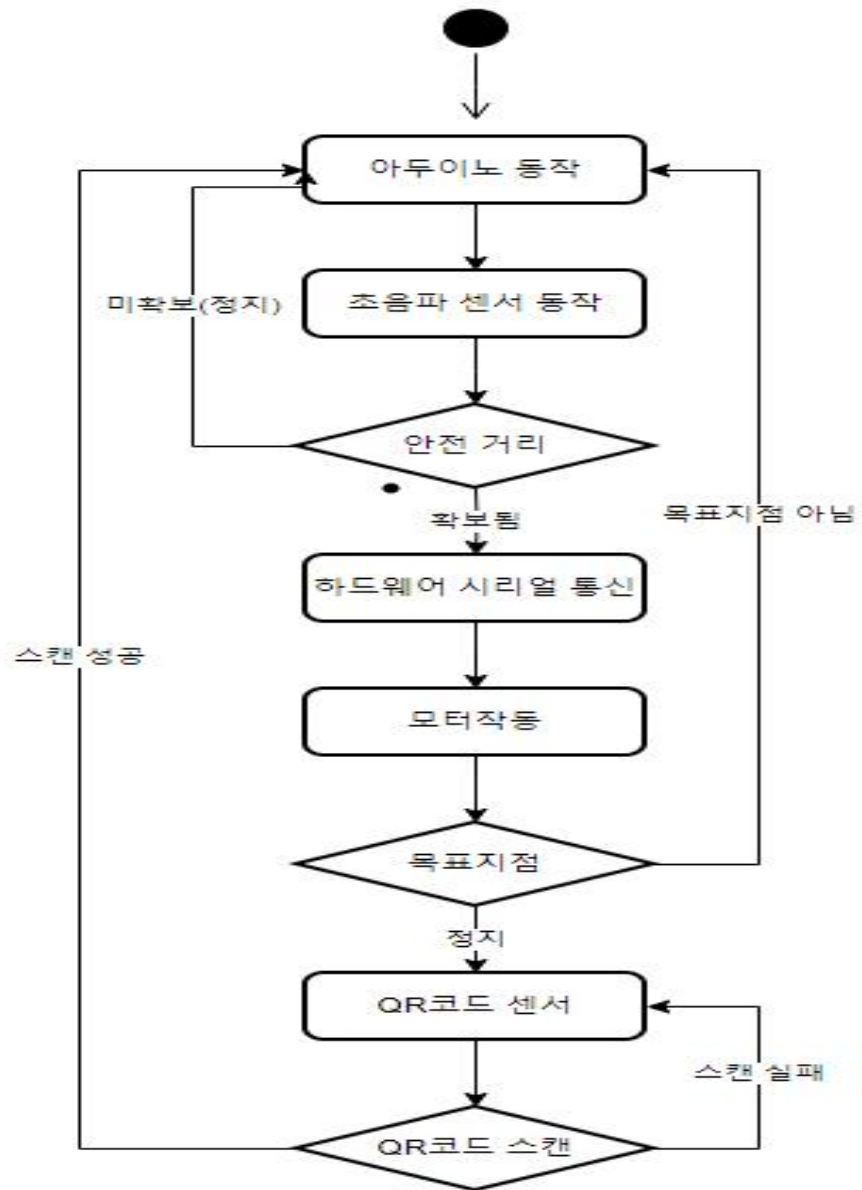
• 사용모듈

- 초음파 센서
- ESP-01(와이파이)
- 적외선센서x3
- DC모터 x4
- L298N
(모터 드라이버)
- GM65
(바코드 카메라)
- UNO x2

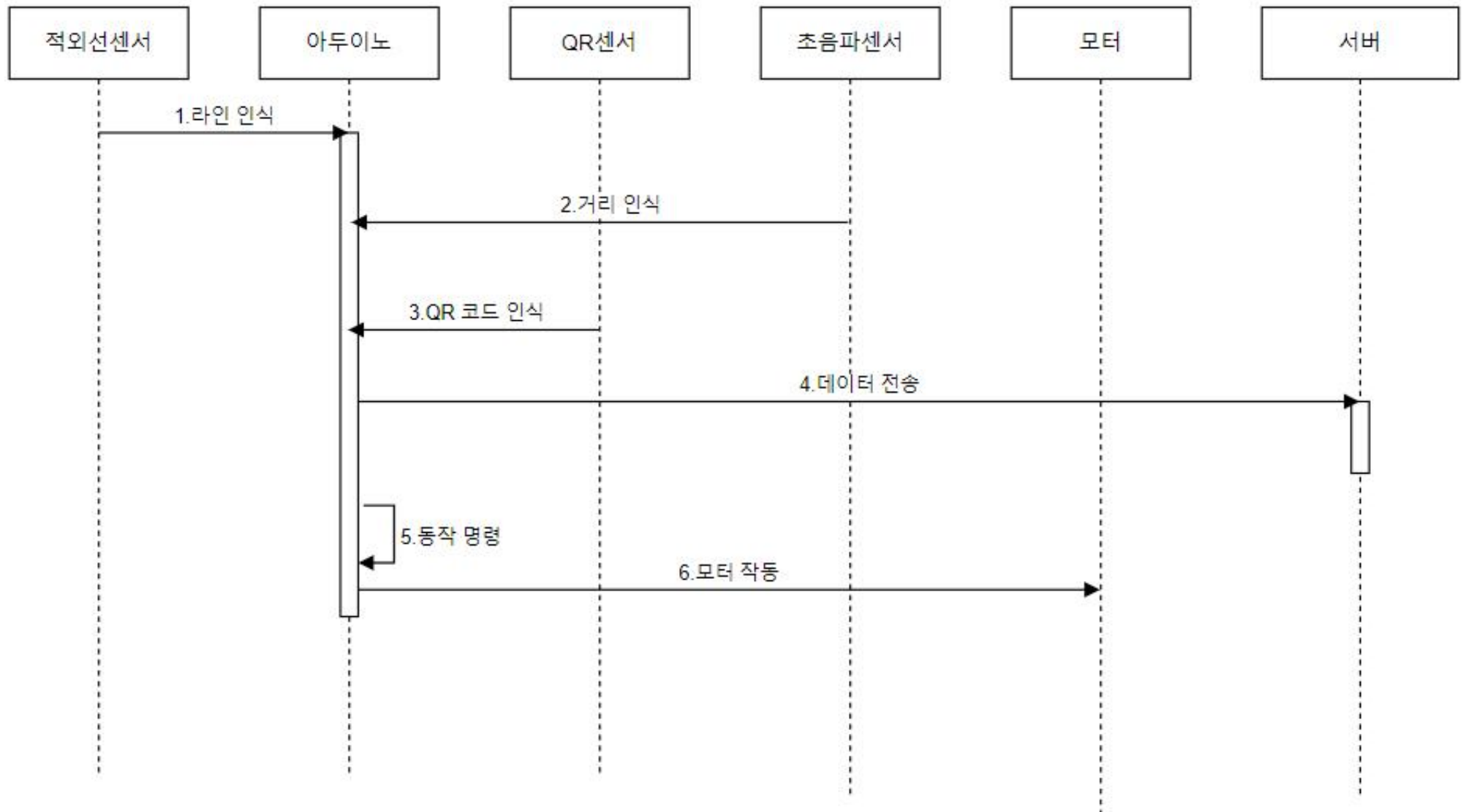
• 설명

- 우노 보드에서는 SoftwareSerial 선언이 1개 이상이 불가능하여 GM65 모듈과 ESP-01모듈을 동시에 사용하기에 어려움이 있음
- 따라서 2개의 우노 보드를 사용, 서로의 보드를 하드웨어 시리얼을 통하여 통신하여 데이터 값을 주고받게 하고 각각의 보드에서 모듈을 사용할 수 있게 함

4. 순서도



5. 시퀀스 다이어그램



6. 코드 명세표

1) 전역변수

모터, 카메라 사용 보드					ESP 사용 보드				
이름	데이터 타입	상수(Y/N)	허용 값	설명	이름	데이터 타입	상수(Y/N)	허용 값	설명
Command	int	Y	1, 2, 3, 4, 5	라인을 따라 움직이게 하는 명령어 변수	data	char	N	-	하드웨어 시리얼로 받은 값을 저장하는 변수
i	int	Y	0, 1	QRcode를 받는 지점과 landing지점을 구분하기 위한 변수	ssid	char	N	-	연결할 WiFi SSID
tempCommand	int	Y	100, 200	화물의 위치를 파악하고 경로를 설정해주는 변수, 값이 100이면 우회전, 200이면 좌회전	pass	char	N	-	연결할 WiFi Password
barcode	String	N	-	하드웨어 시리얼로 보낼 바코드 변수	status		N	-	WiFi 상태 초기값 설정
barcode_Data	String	N	-	GM65모듈로 받은 바코드 입력값	reqCount		N	-	요청 카운터
barcode_char	char	N	-	하드웨어 시리얼로 보내기 위해 char형태로 변환하여 입력 받는 변수	c	char	N	-	클라이언트의 정보
duration	long	N	-	초음파 센서로 받은 음속 변수	-	-	-	-	-
distance	int	N	-	음속을 사용해서 거리 계산값	-	-	-	-	-

2) 함수

모터, 카메라 사용 보드	
이름	설명
forword	전진
turnRight	우회전
turnLeft	좌회전
Stop	전진
right_angle	90도 우회전
left_angle	90도 좌회전
read_sensor_values	로그인 시, id와 pw를 검사하는 함수로 로그인 성공시 1을 반환하고 아니라면 0을 반환 * 설명에 반드시 매개변수 이름과 값을 언급하여 충분히 설명해야함
scanBarcode	GM65 바코드 카메라로 QR코드를 읽는 함수 값을 읽고 send_HW_Serial()을 호출
send_HW_Serial	hardwareSerial통신을 통해 값을 다른 보드로 전송하는 함수, 하드웨어시리얼통신은 char형태만 가능하며 String 형태의 값을 char 형태로 변형하는 코드가 있음
Get_QR_Code_Landing_Point	바코드에서 얻은 값을 바탕으로 방향을 설정하는 함수
landing	도착지점에 도착하면 부저음을 울림
turn_Corner	tempCommand값에 따라 좌우회전을 구분하는 함수
Ultrasonic_Waves	초음파 센서로 일정거리에 물체가 존재하면 자동으로 정지
ESP 사용 보드	
이름	설명
printWifiStatus	연결된 WiFi 상태 출력하는 함수

7. 코드

1) 모터, 카메라 사용 보드(Line_Motor.ino)

<pre>#include <SoftwareSerial.h> SoftwareSerial GM65(2,3); // GM65는 소프트웨어 시리얼을 계속 사용 #define enA 10 // Enable1 L298 Pin enA #define in1 9 // 전진 #define in2 8 // 후진 #define in3 7 // 후진 #define in4 6 // 전진 #define enB 5 // Enable2 L298 Pin enB #define R_SA0 // ir sensor Right #define L_SA1 // ir sensor Left #define Side_SA3 #define piezo 4 // 수동부저 #define trigPin 12 // 초음파 센서의 Trig 핀 #define echoPin 11 // 초음파 센서의 Echo 핀 #define safeDistance 10 // 사람을 감지하는 안전 거리 intCommand = 0; inti=0; inttempCommand = 0; Stringbarcode = ""; longduration; intdistance;</pre>	<pre>voidsetup(){ // put your setup code here, to run once Serial.begin(9600); GM65.begin(9600); pinMode(R_S, INPUT); pinMode(L_S, INPUT); pinMode(Side_S, INPUT); pinMode(enA, OUTPUT); pinMode(in1, OUTPUT); pinMode(in2, OUTPUT); pinMode(in3, OUTPUT); pinMode(in4, OUTPUT); pinMode(enB, OUTPUT); pinMode(trigPin, OUTPUT); pinMode(echoPin, INPUT); pinMode(piezo, OUTPUT); analogWrite(enA, 160); // 모터 세기 조절 analogWrite(enB, 160); // 모터 세기 조절 }</pre>
---	---


```
void forward(){//forward
digitalWrite(in1, HIGH);//Right Motor forward Pin
digitalWrite(in2, LOW);//Right Motor backward Pin
digitalWrite(in3, LOW);//Left Motor backward Pin
digitalWrite(in4, HIGH);//Left Motor forward Pin
}
void turnRight(){//turnRight
digitalWrite(in1, LOW);
digitalWrite(in2, HIGH);
digitalWrite(in3, LOW);
digitalWrite(in4, HIGH);
}
void right_angle(){//turnRight
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, HIGH);
}
void turnLeft(){//turnLeft
digitalWrite(in1, HIGH);
digitalWrite(in2, LOW);
digitalWrite(in3, HIGH);
digitalWrite(in4, LOW);
}
void left_angle(){//turnLeft
digitalWrite(in1, HIGH);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, LOW);
}
void stop(){//stop
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, LOW);
}
```

```
void read_sensor_values()
{
    if(digitalRead(Side_S) == 1) {
        Command = 5;//STOP
    }

    else if((digitalRead(R_S) == 0)&&(digitalRead(L_S) == 0)) {
        Command = 1;// 전진
    }

    else if((digitalRead(R_S) == 1)&&(digitalRead(L_S) == 0)&&(digitalRead(Side_S) == 0)) {
        Command = 2;// 오른쪽으로 라인 맞추기
    }

    else if((digitalRead(R_S) == 0)&&(digitalRead(L_S) == 1)&&(digitalRead(Side_S) == 0)) {
        Command = 3;// 왼쪽으로 라인 맞추기
    }

    else if((digitalRead(R_S) == 1)&&(digitalRead(L_S) == 1)) {
        Command = 4;// STOP
    }
}

void scanBarcode() {
    Serial.println("scanBarcode");
    while(barcode == "") {
        while(GM65.available()) {
            String barcode_Data = GM65.readStringUntil('\n');
            Serial.print("read : ");
            Serial.println(barcode_Data);
            barcode_Data.trim();// trim 함수를 수정하였습니다.
            barcode = barcode_Data;
        }
    }
}
```

```
void send_HW_Serial() {
    char barcode_char = barcode.charAt(0); // String을 char로 변환합니다.
    Serial.write(barcode_char);
    Serial.flush();
    Serial.print("HW_Serial: ");
    Serial.println(barcode_char);
}

void Get_QR_Code_Landing_Point() {
    scanBarcode();

    if (barcode == "a") {
        tempCommand = 100;
    }
    else if (barcode == "b") {
        tempCommand = 200;
    }
}

//부저음 울리기
void landing() {
    Serial.println("Piezo_Ring");
    digitalWrite(piezo, HIGH);
    delay(1000);
    digitalWrite(piezo, LOW);
    delay(1000);
    Serial.println("Piezo_Ring");
}
```

```
void turn_Corner() {
    if (i == 0) {
        if (tempCommand == 100) { //우회전 하는 코드
            right_angle();
            delay(2200);
            forward();
            delay(100);
        }
        else if (tempCommand == 200) { //좌회전 하는 코드
            left_angle();
            delay(2200);
            forward();
            delay(100);
        }
        i++;
        read_sensor_values();
        return;
    }
    else if (i == 1) {
        landing();
        // landing지점 벗어나기
        forward();
        delay(100);

        i = 0;
        barcode = "";
        read_sensor_values();
        return;
    }
}
```

```
void Ultrasonic_Waves() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.0342; // 음속을 사용해서 거리 계산
  Serial.print("Distance: ");
  Serial.println(distance);

  if(distance < safeDistance) {
    Command = 4;
  }
}
```

```
void loop(){
  read_sensor_values();
  Ultrasonic_Waves();
  send_HW_Serial();

  if(Command == 1){
    forward();
    Serial.println("command1");
  }
  if(Command == 2){
    turnRight();
    Serial.println("command2");
  }
  if(Command == 3){
    turnLeft();
    Serial.println("command3");
  }
  if(Command == 4){
    Stop();

    Serial.println("command4");
  }
  if(Command == 5){
    Serial.println("command5");
    Stop();
    Get_QR_Code_Landing_Point();
    turn_Corner();
  }
}
```

2) ESP 사용 보드(ESP_Server_Final.ino)

```
#include "WiFiEsp.h"// ESP8266 WiFi 모듈을 위한 라이브러리
#ifndef HAVE_HWSERIAL1// 하드웨어 시리얼이 없는 경우에 대비한 조건문
#include "SoftwareSerial.h"// 소프트웨어 시리얼 라이브러리
SoftwareSerial ESP(2, 3);// RX, TX
#endif
#include <stdio.h>
char data;// 데이터를 저장할 char 타입 변수 선언
char databox;
const char ssid[] = "KT_GiGA_2G_Wave2_BDB2";// 연결할 WiFi SSID
const char pass[] = "bf4edc0603";// 연결할 WiFi Password
int status = WL_IDLE_STATUS;// WiFi 상태 초기값 설정
int reqCount = 0;// 요청 카운터
WiFiEspServer server(80);// 80 포트로 서버 시작

void setup(){
  Serial.begin(9600);// 시리얼 통신 시작
  ESP.begin(9600);// ESP8266 시리얼 통신 시작
  WiFi.init(&ESP);// WiFi 초기화
  if(WiFi.status() == WL_NO_SHIELD) {// WiFi 쉴드가 없는 경우 에러
    메시지 출력
    Serial.println("WiFi shield not present");
    while(true);
  }
  while( status != WL_CONNECTED) {// WiFi에 연결될 때까지 반복
    Serial.print("Attempting to connect to WPA SSID: ");
    Serial.println(ssid);
    status = WiFi.begin(ssid, pass);// WiFi 연결 시도
  }
  Serial.println("You're connected to the network");// 연결 성공 메
  시지 출력
  printWifiStatus();// WiFi 상태 출력
  server.begin();// 서버 시작
}
```

```
void loop() {
  // 들어오는 클라이언트 체크
  WiFiEspClient client = server.available();
  // 시리얼에서 데이터 읽고 'data'에 저장
  if(Serial.available()) {
    data = Serial.read();
    Serial.println(data);

    if(data == 'a') {
      databox = data;
    }
    else if(data == 'b') {
      databox = data;
    }
    Serial.println("-----");
    ;
    Serial.print("Data_Print: ");
    Serial.println(databox);// 읽은 데이터 출력
    Serial.println("-----");
    ;
  }
  if(client) {// 클라이언트가 연결된 경우
    Serial.println("New client");
    boolean currentLineIsBlank = true;
```

```

while(client.connected()) {// 클라이언트가 연결된 동안
    if(client.available()) {// 클라이언트로부터 데이터가 도착한 경
우
        charc = client.read();// 데이터 읽기
        Serial.write(c);// 읽은 데이터 쓰기
        if(c == '\n' && currentLineIsBlank) {
            Serial.println("Sending response");
            ,,,
            client.print("Received data: ");
            client.print(databox);// 시리얼에서 읽은 데이터를 웹 페이지에 출력합
            니다.
            client.print("<br>\r\n");

            client.print("-----");
            client.print("<br>\r\n");
            client.print("Container: a");
            client.print("<br>\r\n");
            client.print("Contents: Galaxy");
            client.print("<br>\r\n");
            client.print("Location: Right");
            client.print("<br>\r\n");
            ,,,
            break;
        }
    }
}

```

```

        if(c == '\n') {
            currentLineIsBlank = true;
        }
        else if(c != '\r') {
            currentLineIsBlank = false;
        }
    }
}
// 웹 브라우저가 데이터를 받는 시간을 주기 위해 10ms 대기
delay(10);
// 연결 종료
client.stop();
Serial.println("Client disconnected");// 클라이언트 연결 종료 메시
지 출력
}
}

void printWifiStatus() {
    // 연결된 네트워크의 SSID 출력
    Serial.print("SSID: ");
    Serial.println(WiFi.SSID());
    // WiFi 쉼드의 IP 주소 출력
    IPAddress ip = WiFi.localIP();
    Serial.print("IP Address: ");
    Serial.println(ip);
    // 브라우저에서 이 페이지를 보려면 어디로 가야 하는지 출력
    Serial.println();
    Serial.print("To see this page in action, open a browser to
    http://");
    Serial.println(ip);
    Serial.println();
}
}

```

8. 역할분담

이름	참여도(%)	내용	비고
윤건용	50	<ul style="list-style-type: none"> • 라인트레이서 기능 구현 • GM65 기능 접합 • 통신 기능 접합 • 모듈 제작 	
이호규	30	<ul style="list-style-type: none"> • 통신 기능 구현 • GM65 함수 구현 • 발표자료 제작 	
황수민	20	<ul style="list-style-type: none"> • 순서도, 다이어그램 제작 • 초음파 센서(거리측정) 	