

# 아두이노 프로젝트

# 목차

## 1.서론

### 1.1 장치 개요

### 1.2 아두이노 프로그램

## 2. 본론

### 2.1 입력장치와 출력장치

### 2.2 핀 배치도

#### 2.2.1 RC Car 핀 배치도

#### 2.2.2 주차장 관리소 핀 배치도

### 2.3 주차장 관리소 작동원리

### 2.4 프로그램 구현-주차장 관리소 코딩 소스

### 2.5 RC Car 작동원리

### 2.6 프로그램 구현-RC Car 코딩 소스

## 3. 결론

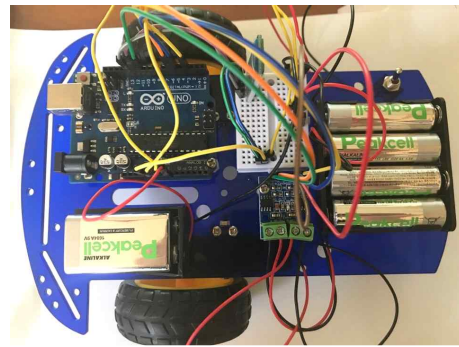
### 3.1 결과(최종 작동원리)

### 3.2 느낀점

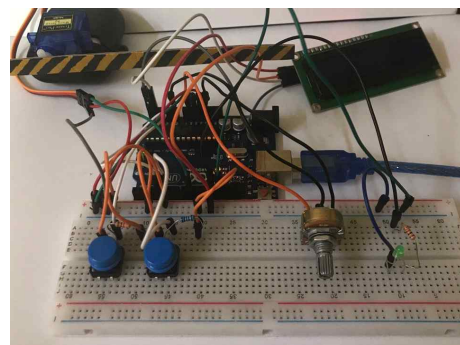
# 1. 서론

## 1.1 장치 개요

장치	부품
RC Car	아크릴 프레임
	모터
	캐스터
	타이어
	토글 스위치
	모터 드라이버(L9110S)
	BREAD BOARD
	AA 배터리 홀더
	9V 배터리 홀더
	JUMPER CABLE
	USB CABLE
	UNO
	AA배터리
	9V배터리
	블루투스 모듈
	볼트와 너트
	모터 브라켓

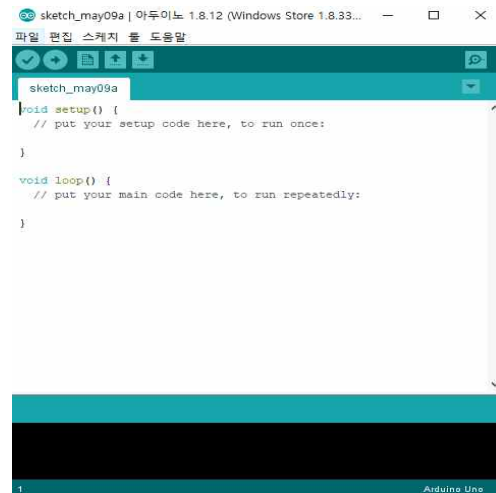


장치	부품
주차장 관리소	UNO R3
	USB CABLE
	JUMPER WIRE
	BREAD BOARD
	CABLE
	POTEN-TIOMETER
	SERVO
	SWITCHES WITH CAPS
	LED
	RESISTANCE
	LCD



## 1.2 아두이노 프로그램

IDE: 효율적으로 소프트웨어를 개발하기 위한 통합개발환경 소프트웨어 어플리케이션 인터페이스이다. 코드 편집기, 디버거, 컴파일러, 인터프리터 등을 포함하고 개발자에게 제공한다. 코드 작성을 위한 텍스트 편집기, 메시지 영역, 텍스트 콘솔, 공통 기능을 위한 버튼이 있는 도구 모음, 일련의 메뉴가 있다. 그것은 프로그램을 업로드하고 그들과 통신하기 위해 아두이노와 Genuino 하드웨어에 연결된다.



## 2. 본론

### 2.1 입력 장치와 출력장치

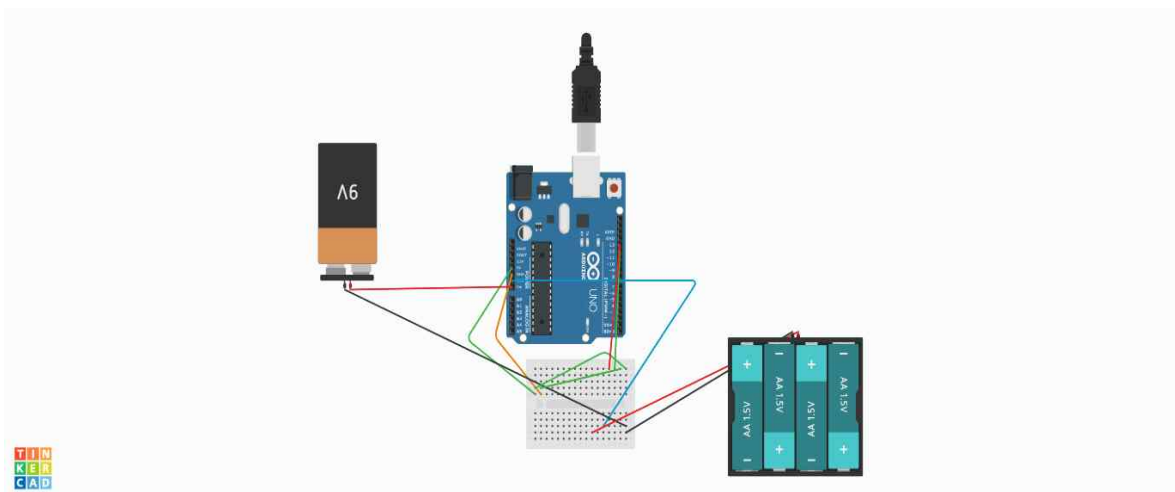
	입력 장치	출력장치
RC Car	토글 스위치	바퀴
	블루투스 모듈	바퀴
주차장 관리소	스위치(버튼1 )	서보모터 (주차장 차단기)
	스위치(버튼2 )	LCD (환영인사)
	가변저항(POTEN-TIOMETER)	LED (차가 들어옴)

## 2.2 핀 배치도

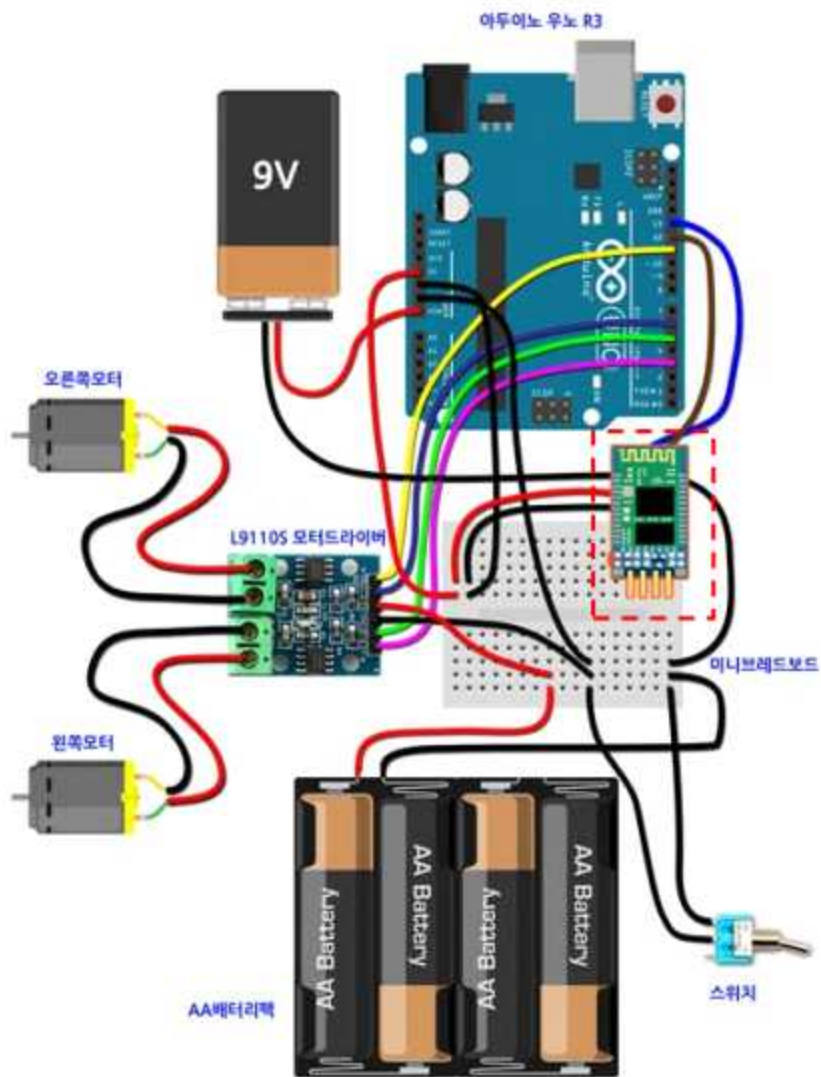
### 2.2.1 RC Car 핀 배치도

1. 스위치선 중 1개를 미니브레드보드 아래쪽 1번째라인에 연결한다
2. 스위치선 중 나머지 1개를 아래쪽 7번째라인에 연결한다
3. 배터리팩에서 나오는 GND를 아래쪽 1번째 라인에 연결한다
4. 점퍼선을 이용해 브레드보드 아래쪽 7번째라인과  
L9110S모터드라이버에 GND를 연결 한다
5. 점퍼선을 이용해 브레드보드 아래쪽 7번째라인과 아두이노 우노 R3에 GND에  
연결한다
6. 스위치선 중 1개를 미니브레드보드 아래쪽 1번째라인에 연결한다
7. 스위치선 중 나머지 1개를 아래쪽 7번째라인에 연결한다
8. 배터리팩에서 나오는 GND를 아래쪽 1번째 라인에 연결한다
9. 점퍼선을 이용해 브레드보드 아래쪽 7번째라인과  
L9110S모터드라이버에 GND를 연결 한다
10. 점퍼선을 이용해 브레드보드 아래쪽 7번째라인과 아두이노 우노 R3에  
GND에 연결한다
11. 배터리팩에서 나오는 VCC를 아래쪽 10번째 라인에 연결한다
12. 점퍼선을 이용해 브레드보드 아래쪽 10번째라인과  
L9110S모터드라이버에 VCC를 연결 한다
13. 왼쪽모터에 나와있는 선을 L9110S 아래쪽 VCC와GND에 연결한다
14. 오른쪽모터에 나와있는 선을 L9110S 위쪽 VCC와GND에 연결한다  
[모터제어신호 점퍼선 연결]
15. 모터드라이버 L9110S 모듈 `B-IA` -> 아두이노 우노 3번 핀에 연결한다
16. 모터드라이버 L9110S 모듈 `B-IB` -> 아두이노 우노 5번 핀에 연결한다

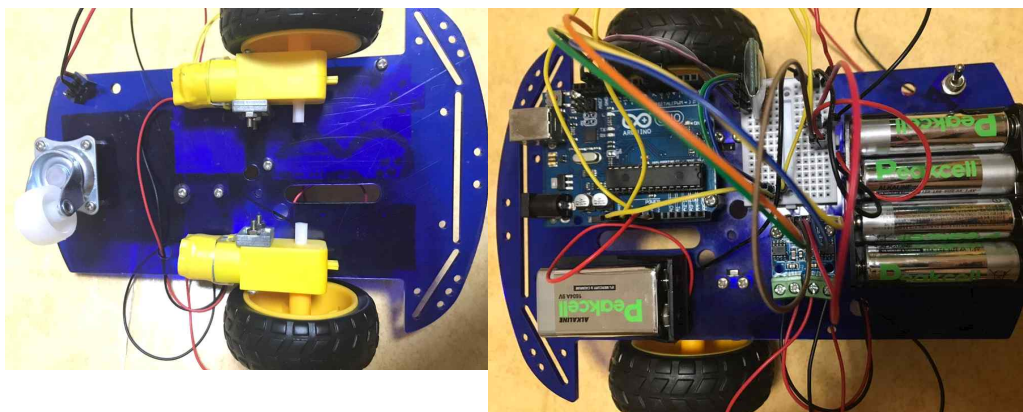
17. 모터드라이버 L9110S 모듈 'A-IA' -> 아두이노 우노 6번 핀에 연결한다
18. 모터드라이버 L9110S 모듈 'A-IB' -> 아두이노 우노 11번 핀에 연결한다
19. 9V용 배터리홀더 GND선 -> 미니 브레드보드 아래쪽 1번째 라인에 연결한다  
(스위치 하나로 아두이노와 모터를 ON/OFF 하기 위해)
20. 9V용 배터리홀더 VCC선 -> 아두이노 우노 R3 보드에 'VIN'에 연결한다
21. 아두이노 우노 R3 5V용 VCC선 -> 미니브레드보드 위쪽 17번째 라인에  
연결한다
22. 아두이노 우노 R3 GND선 -> 미니브레드보드 위쪽 16번째 라인에 연결한다
23. 미니 브레드보드 위쪽 17번째 라인 ->  
미니 브레드보드 위쪽 1번째 라인에 연결한다(VCC)
24. 미니 브레드보드 위쪽 16번째 라인 ->  
미니브레드보드 위쪽 2번째 라인에 연결한다(GND)
25. 아두이노 보드에서 12번PIN -> 미니브레드보드 위쪽 3번째 라인에 연결한다  
(TX)
26. 아두이노 보드에서 13번PIN -> 미니브레드보드 위쪽 4번째 라인에 연결한다  
(RX)
27. 블루투스모듈을 미니브레드보드에 연결 한다



**AUTODESK TINKERCAD** 사이트를 이용하여 RC Car 핀 배치도를 만들었다



[ 전체적 RC Car 배치도 ]



[ 실제 RC Car 배치도 사진 ]

## 2.2.2 주차장 관리소

### 1) 텍스트스위치 버튼

텍스트스위치 버튼1은 주차장 차단기연결이고 텍스트 스위치 버튼2는 LCD (환영 인사)이다.

서보모터는 9번 핀에 텍스트 스위치(버튼)은 4,5번핀에 저항과 텍스트 스위치(버튼) 있는 곳에 점퍼선을 연결하고 VCC와 GND는 브레드 보드에 +,-에 연결하여 점퍼선을 이용해 아두이노 5V, GND에 연결한다.

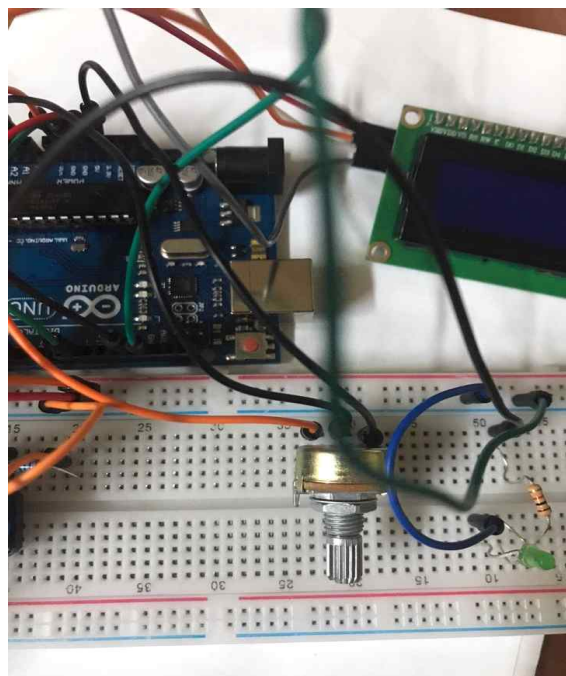
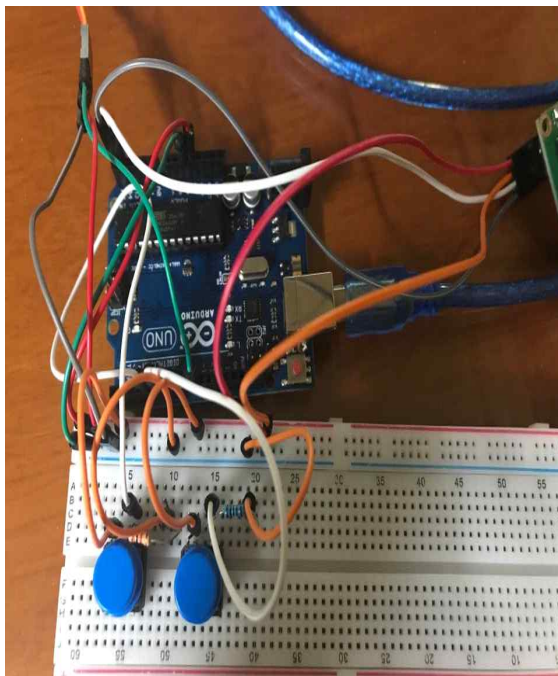
버튼2 - LCD는 4개의 점퍼선 중 2개는 A4, A5에 연결하고 나머지 2개는 브레드 보드에 연결한다.

### 2) 가변저항

브레드보드에 가변저항 POTENTIOMETER를 연결한다

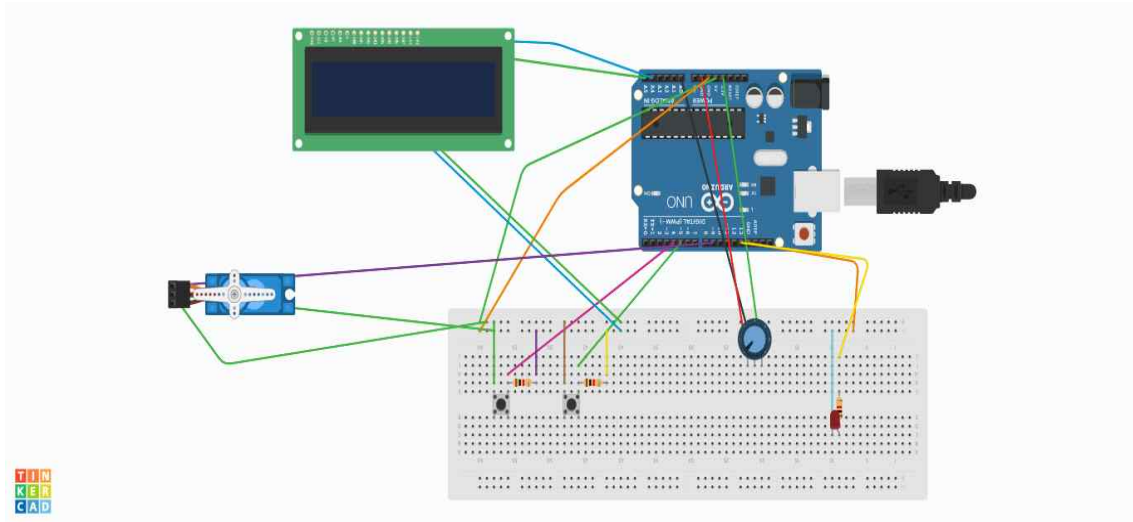
그 앞에 점퍼선을 3개를 연결하여 1개는 GND , 1개는 A0 ,1개는 3.3V에 연결한다

또한 전구와 저항을 연결한다



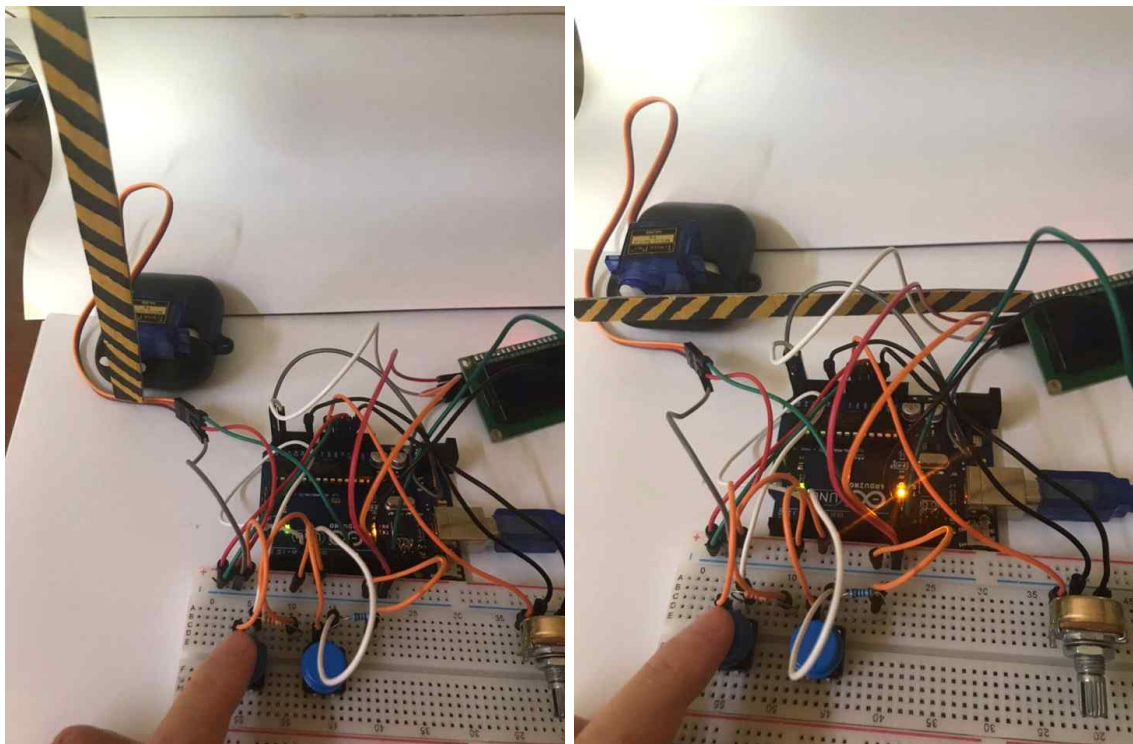
[실제 배치도 사진]





AUTODESK TINKERCAD 사이트를 이용하여 주차관리소 핀 배치도를 만들었다

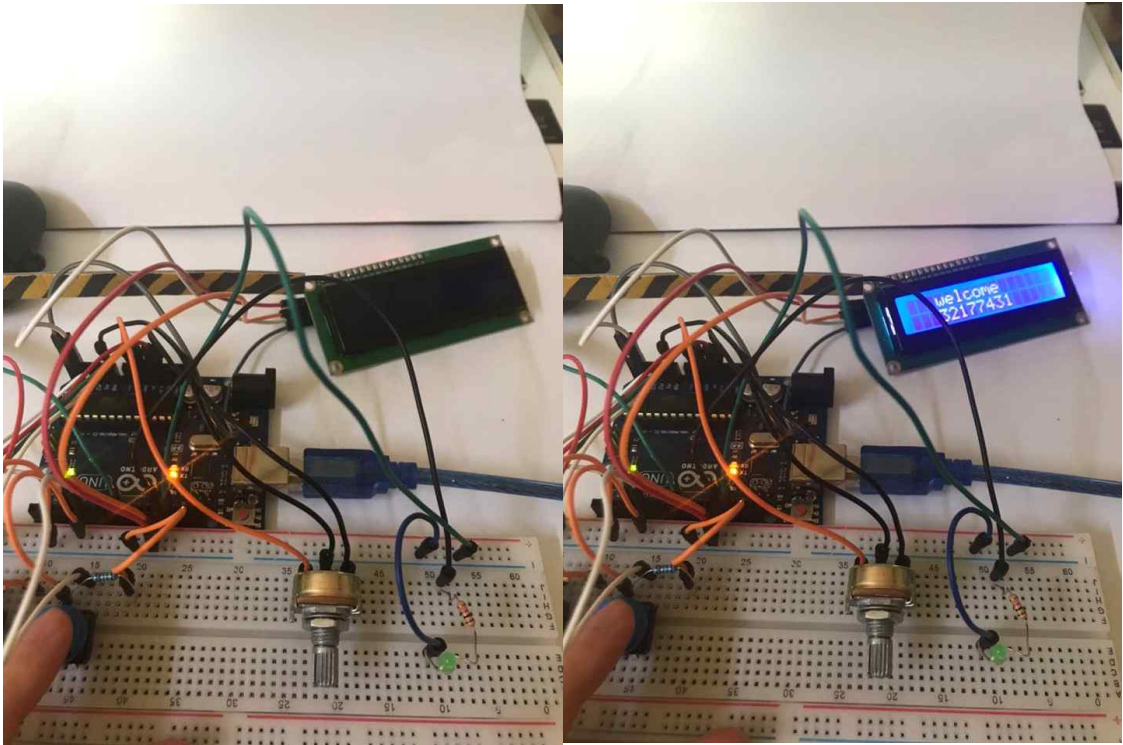
## 2.3 작동원리



[주차장 차단기]

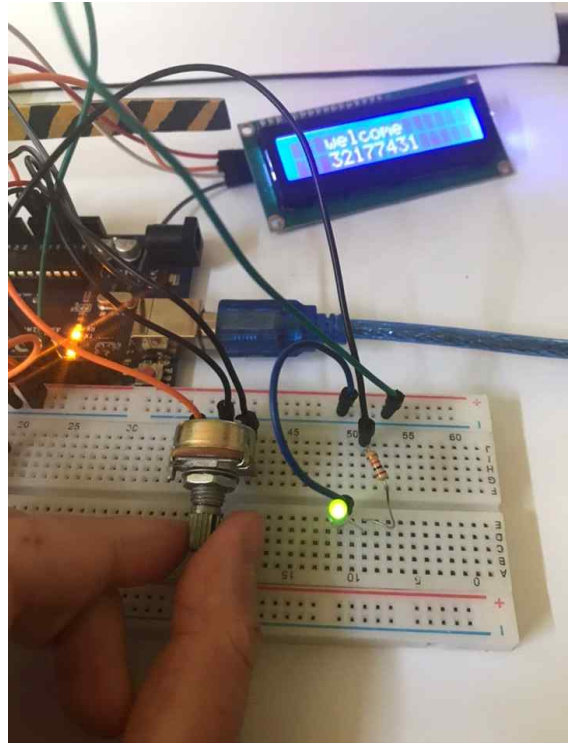
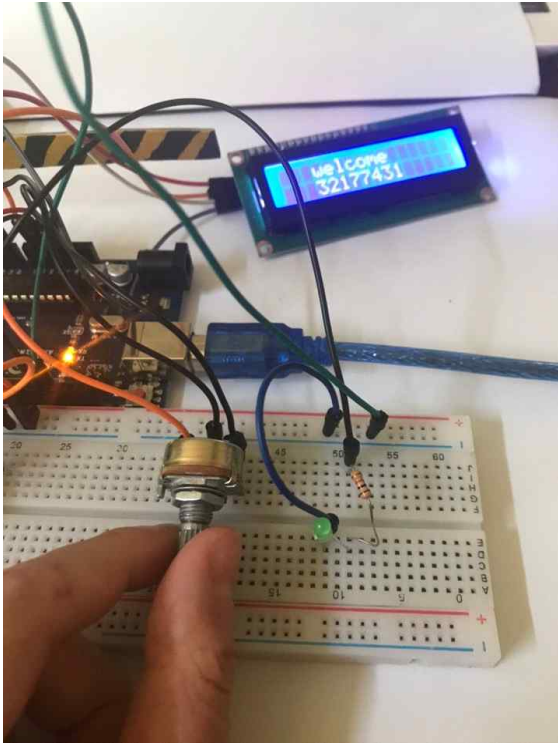
위의 사진은 텍스트스위치(버튼1)을 눌렀을 때 사진이다

왼쪽 사진은 텍스트스위치 (버튼1)을 눌렀을 때 5초간 주차장 차단기가 90도로 열린다



[ lcd 환영 인사]

위 사진은 텍스트スイッチ 버튼2(LCD 환영인사)이다.  
버튼을 누르면 화면이 켜져 'welcome 32177431' 이 보인다  
또한 버튼을 다시 누르면 꺼진다.



[ 가변저항 ]

위 사진은 가변저항을 돌렸을 때 led에 불이 켜지는 사진이다

## 2.4 프로그램 구현-주차장 관리소 코딩소스

```
#include<Servo.h>

#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27,20,4);    // Set the LCD adress to 0x27 for a 16 chars and
                                     2line display

Servo myservo;                      // Servo를 myservo로 정의
int button1 = 4;                    // 버튼 1번을 4번핀으로 설정
int button2 = 5;                    // 버튼 1번을 5번핀으로 설정

void setup(){
  Serial.begin(9600);

  pinMode(13, OUTPUT);

  lcd.init();
  lcd.init();

                                     // print a message to LCD
  lcd.backlight();                  // LCD backlight을 켜
  lcd.setCursor(3,0);               // (3,0) 좌표로 커서를 위치
  lcd.print("welcome");
  lcd.setCursor(3,1);               // (3,1)의 좌표로 커서를 위치
  lcd.print("32177431");

  myservo.attach(9);                // 서보모터 9번핀으로 설정
  pinMode(button1,INPUT);            // 텍스트스위치 1번 INPUT으로 설정
  pinMode(button2,INPUT);            // 텍스트스위치 2번 INPUT으로 설정
}

void loop(){
  int A(map(analogRead(A0) , 0, 1023, 0, 1000));
  Serial.println(A) ;
  if(A>50) {
                                     // 50초과일 때 불이 들어옴
    digitalWrite(13,1);
  }
}
```

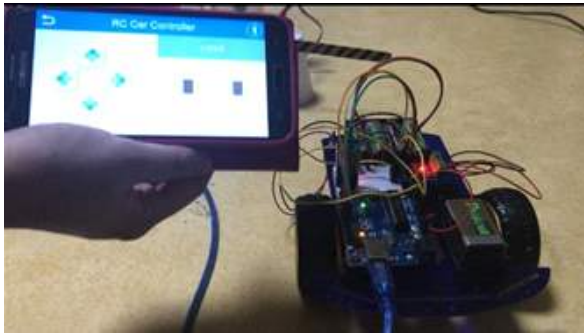
```

else if(A<=50) {
    digitalWrite(13,0);
} // 50이하일 때 불이 꺼짐

myservo.write(90);
if(digitalRead(button1)==HIGH){
    myservo.write(90); // 90일 때 멈춤
    myservo.write(180);
    delay(10000); // 10초 동안 차단기 열린 상태 유지
}
if(digitalRead(button2)==HIGH){
    lcd.noBacklight(); // 만약 버튼2를 누르면 LCD backlight을 끄
}
if(digitalRead(button2)==HIGH){ // 만약 버튼 2를 누르면 LCD backlight을 켜
    lcd.backlight();
}
}

```

## 2.5 RC Car 작동원리



위 사진은 RC카를 작동하는 것이다  
안드로이드 폰에 CODING CONNECT 라는 앱을 다운로드 한다  
RC카에는 블루투스 모듈을 미니 브레드 보드에 연결한다  
앱에 RC 카 블루투스 모듈하고 연결하여 조종이 가능하다

## 2.6 프로그램 구현-RC Car 코딩

```
#include <SoftwareSerial.h>
int A_1A = 6;
int A_1B = 11;
int B_1A = 3;
int B_1B = 5;                // 오른쪽 모터 L9110s A_1A 6, L9110s A_1B 11
                             // 왼쪽 모터 L9110s B_1A 3, L9110s B_1B 5

int btRxBPin = 12;
int btTxPin = 13;           //블루투스 모듈

SoftwareSerial bluetoothSerial(btRxBPin, btTxPin);
int speed = 150;

void setup() {
    Serial.begin(9600);
                                // 블루투스를 baud rate 9600으로 시작

    bluetoothSerial.begin(9600);
                                // 핀을 초기화
                                // L9110S 모터드라이버의 핀들을 출력으로 변경

    pinMode(A_1A, OUTPUT);
    pinMode(A_1B, OUTPUT);
    pinMode(B_1A, OUTPUT);
    pinMode(B_1B, OUTPUT);
    digitalWrite(A_1A, LOW);
    digitalWrite(A_1B, LOW);
    digitalWrite(B_1A, LOW);
    digitalWrite(B_1B, LOW);
}
void loop() {

    char cmd ;

    if (bluetoothSerial.available()) {
        cmd = bluetoothSerial.read();
        Serial.print(cmd);

        switch (cmd) {
            case 'f':                // 'f'는 전진 명령
                //모터A 정회전
                analogWrite(A_1A, speed);
```

```

    analogWrite(A_1B, 0);
    //모터B 정회전
    analogWrite(B_1A, speed);
    analogWrite(B_1B, 0);
    break;
case 'b':                                // 'b'는 후진 명령
    //모터A 역회전
    analogWrite(A_1A, 0);
    analogWrite(A_1B, speed);
    //모터B 역회전
    analogWrite(B_1A, 0);
    analogWrite(B_1B, speed);
    break;
case 'r':                                // 'r'은 오른쪽 명령
    //모터A 정회전
    analogWrite(A_1A, 0);
    analogWrite(A_1B, 0);
    //모터B 역회전
    analogWrite(B_1A, speed);
    analogWrite(B_1B, 0);
    break;
case 'l':                                // 'l'은 왼쪽 명령
    //모터A 역회전
    analogWrite(A_1A, speed);
    analogWrite(A_1B, 0);
    //모터B 정회전
    analogWrite(B_1A, 0);
    analogWrite(B_1B, 0);
    break;
case 's':                                // 's'는 정지 명령
    //모터A 정지
    analogWrite(A_1A, 0);
    analogWrite(A_1B, 0);
    //모터B 정지
    analogWrite(B_1A, 0);
    analogWrite(B_1B, 0);
    break;
}
}
}

```



### 3. 결론

#### 3.1 결과(최종 작동원리)

블루투스 연결된 폰으로 작동하여 RC Car를 움직인다.

RC Car가 주차장 차단기에 가까이 오면 LCD가 켜지고 환영 인사가 켜진다.

그리고 주차장 차단기가 10초간 올라가며 RC Car가 지나가면 가변저항을 통해 램프에 불이 켜진다.

#### 3.2 느낀점

RC카 키트를 구매하여 주차장 관리시스템을 만들 수 있었다.

기존에 있는 키트에 주차장 관리시스템을 만들기에는 부품이 부족하였지만 따로 부품을 구매하여 완성도를 높였다.

여러 센서를 이용하여 주차장 차단기에 가까이 왔을 경우 인식하여 저절로 차단기가 열리던가 차가 들어온 후 주차를 할 경우 터치 센서를 이용하여 불이 들어온다는 것을 더 나아가 생각하게 되었다.