



오토믹 자동차

Android 앱을 이용한 주행

서강원
김기영
김채진
옥진해
전고은

Contents



01

목표

02

회로 연결 & 추가 변동 사항

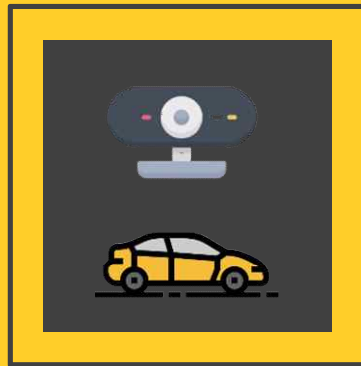
03

Applicat ion

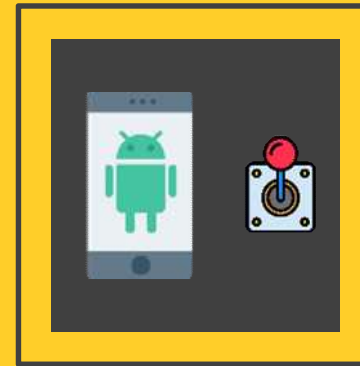
04

DEMO

10



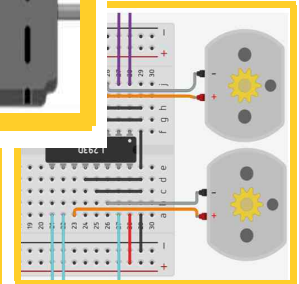
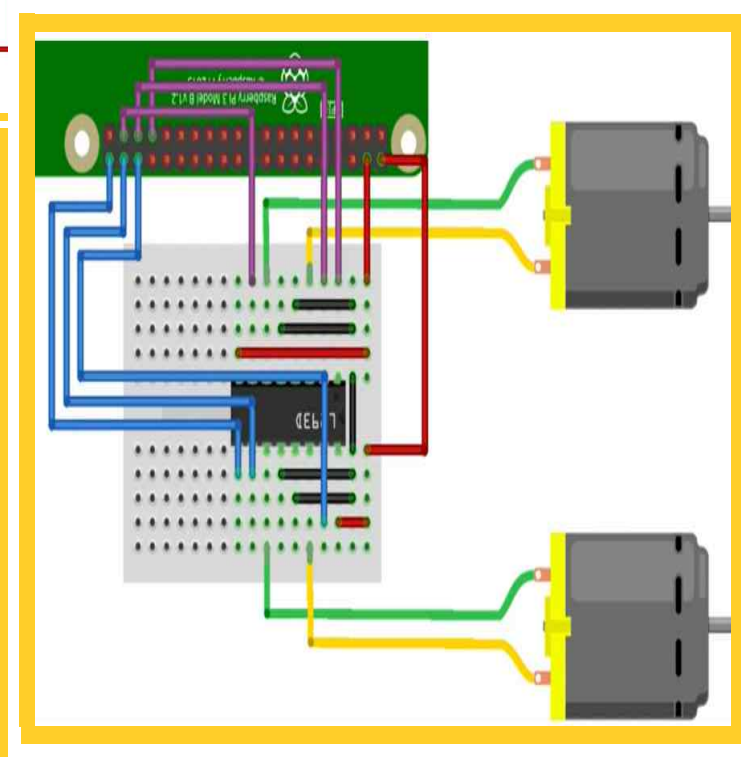
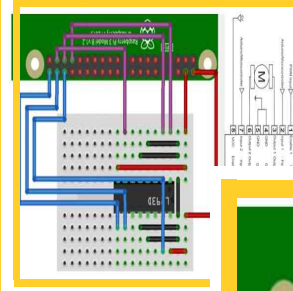
Line Tracking



Joystick Controller

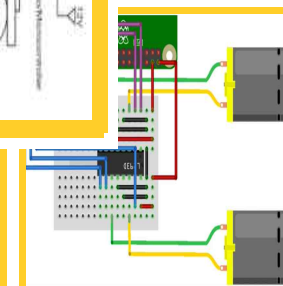
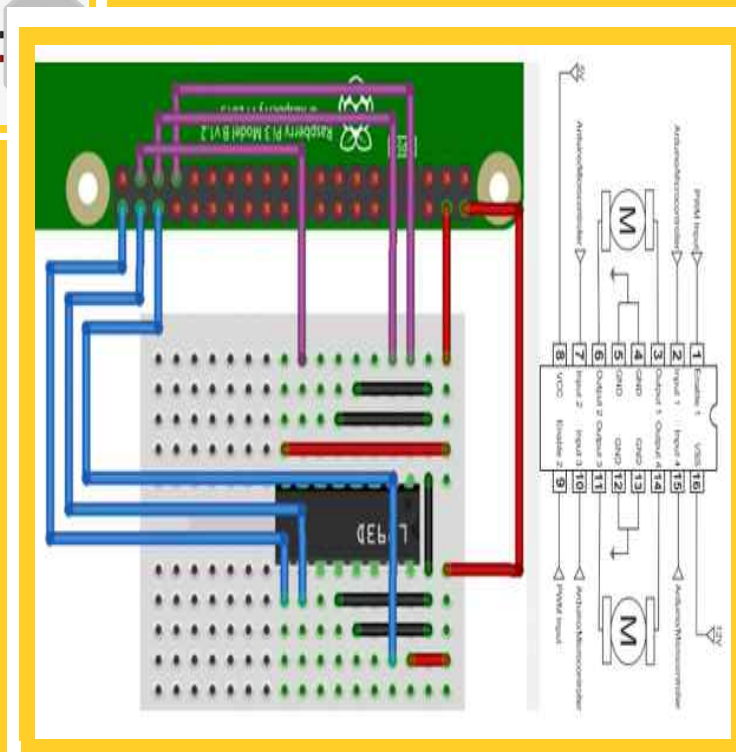
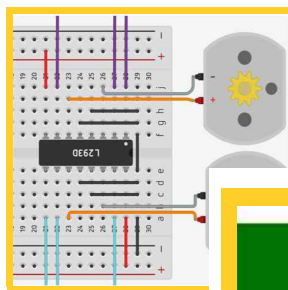


회로 연결 과정



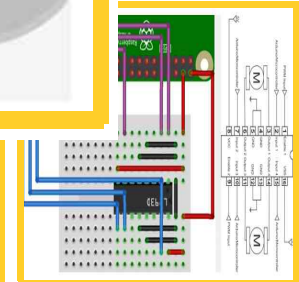
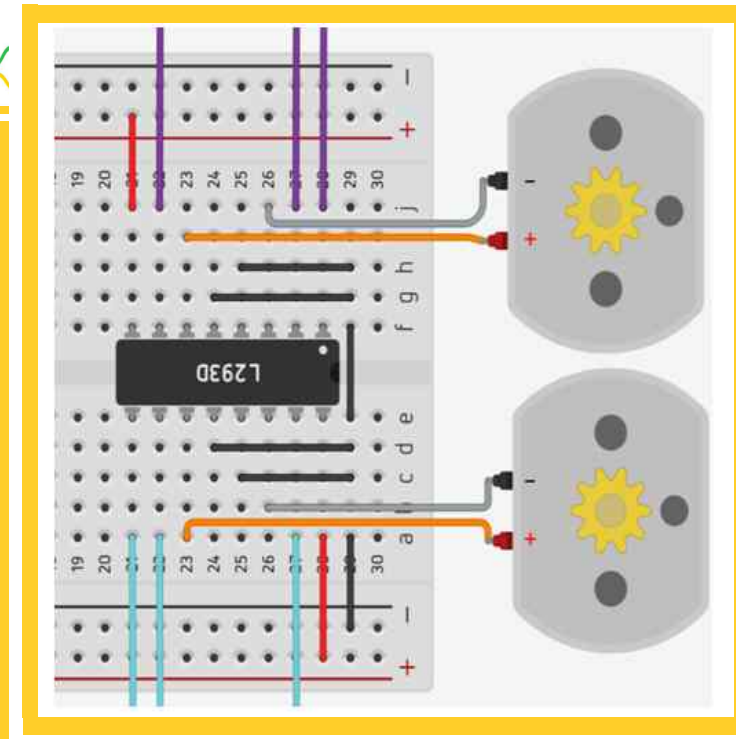
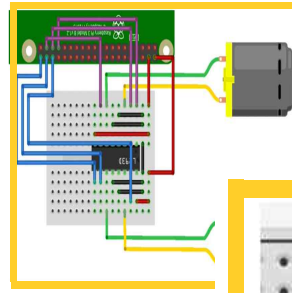
회로 연결 과정

| Rpi | L293D | Direct ion |
|--------|-----------|----------------|
| 5V | 8,16 | |
| GND | 4,5,12,13 | |
| GPIO13 | 1 | LEFT_PWM |
| GPIO19 | 2 | LEFT_FORWARD |
| GPIO26 | 7 | LEFT_BACKWARD |
| GPIO21 | 9 | RIGHT_PWM |
| GPIO16 | 15 | RIGHT_FORWARD |
| GPIO20 | 10 | RIGHT_BACKWARD |

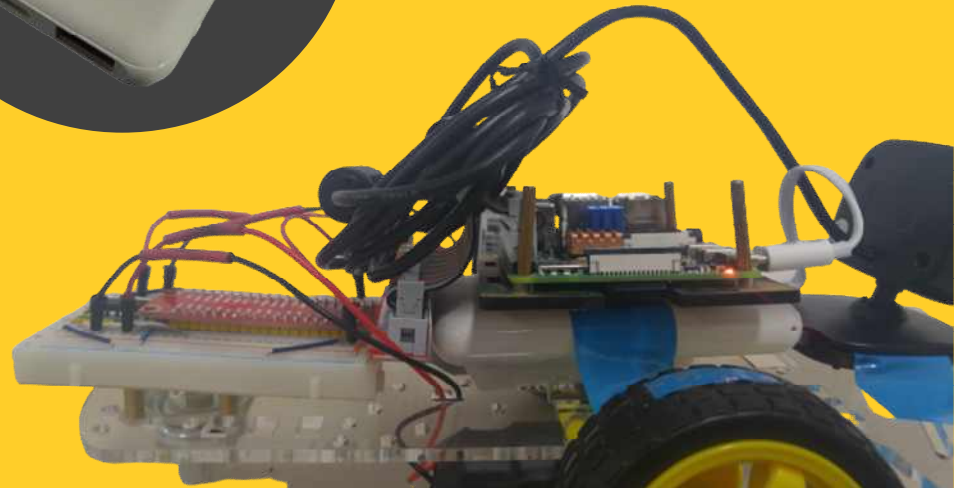
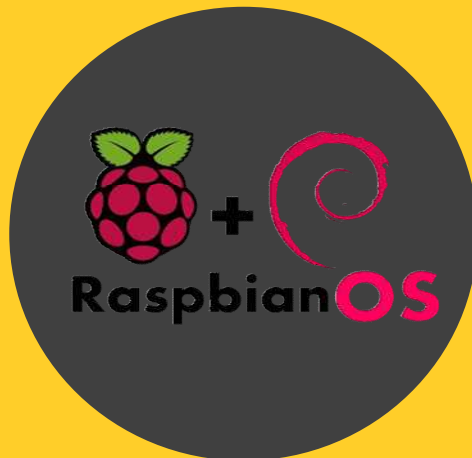


회로 연결 과정

| Rpi | L293D | Mot or |
|-----|-------|------------|
| 5V | 6 | 5V(Orange) |
| GND | 3 | GND(Gray) |
| 5V | 11 | 5V(Orange) |
| GND | 14 | GND(Gray) |

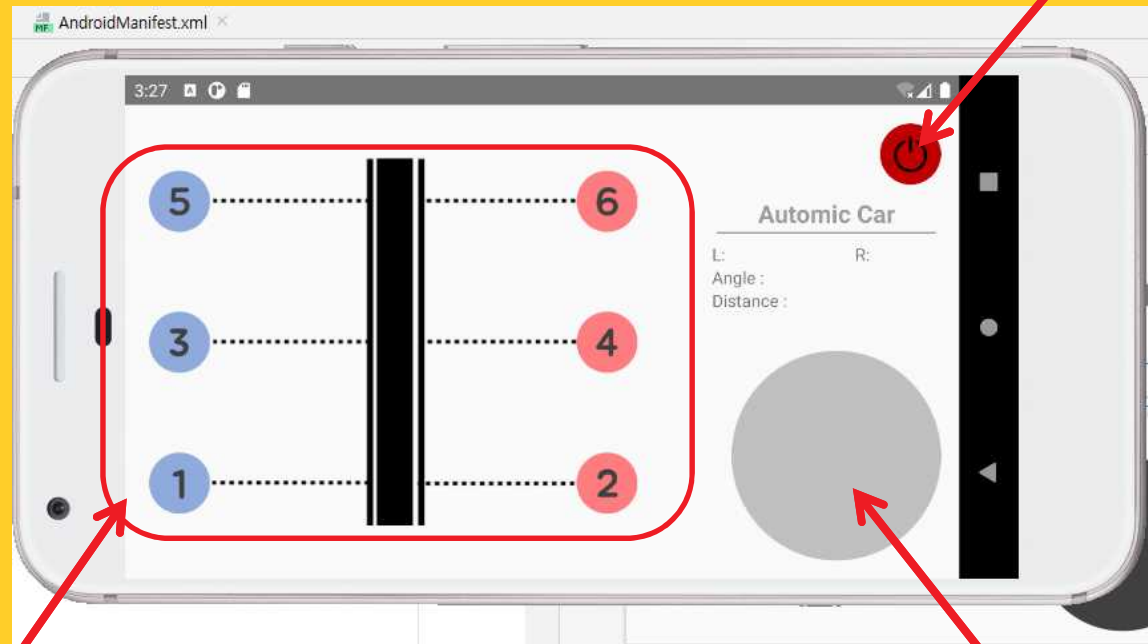


추가 변동 사항



Applicat ion

종료 버튼 (실행 종료)

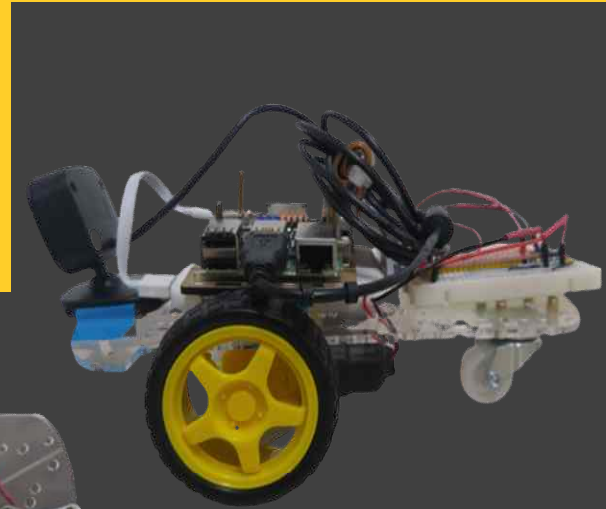
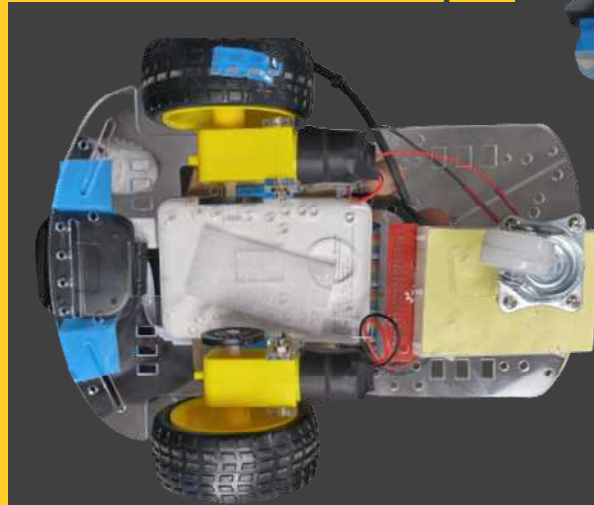


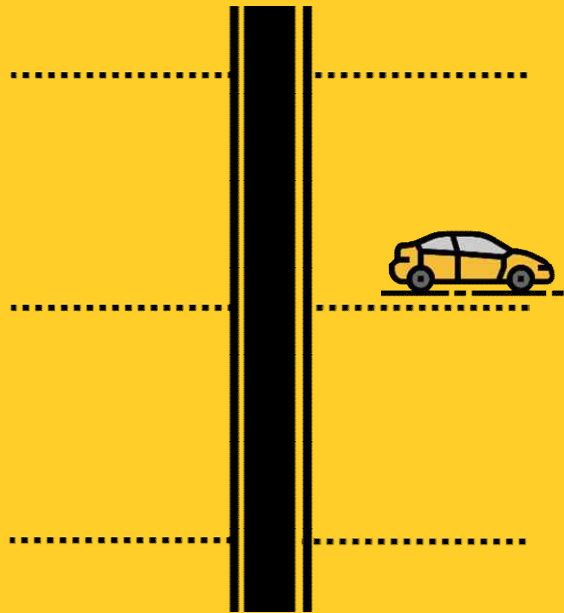
Line Tracking 버튼

Joyst ick Cont r oller

힘들었던 점(H/W)

- i) 모터드라이버
 - 성능에 따른 불균형
- ii) 무게중심
 - 뒷 바퀴 제어
- iii) 카메라 영상 처리
 - 카운트



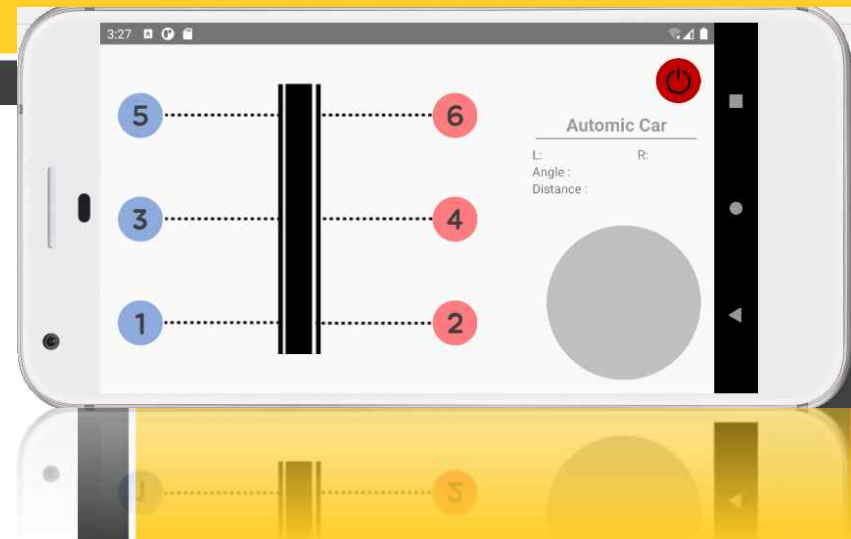


Line Tracking

Line Tracking



```
btn1.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        mSendData = new SendData();
        CMD = "1,_";
        editText.setText("운행중");
        // 보내기 시작
        mSendData.start();
        btnCancel.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                dialog.dismiss();
            }
        });
        dialog.show();
    }
});
```



Line Tracking



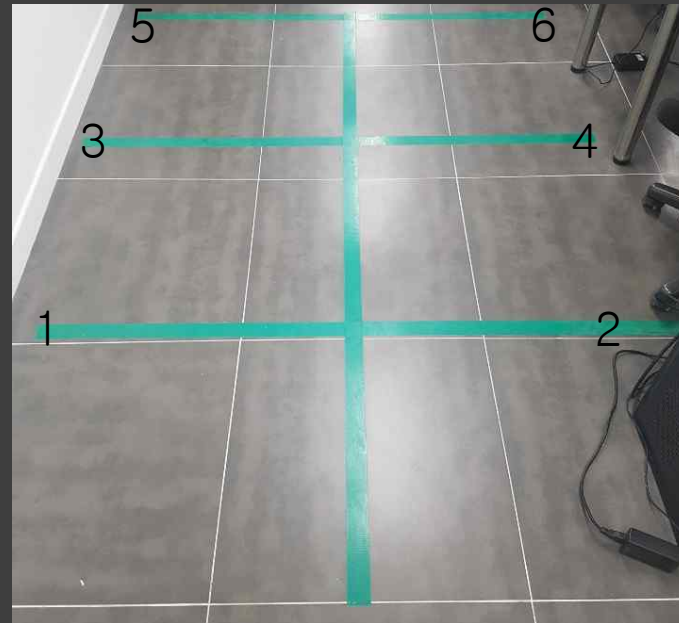
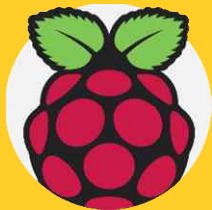
```
btn1.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        mSendData = new SendData();
        CMD = "1,_";
        editText.setText("운행중");
        // 보내기 시작
        mSendData.start();
        btnCancel.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                dialog.dismiss();
            }
        });
        dialog.show();
    }
});
```



Line Tracking



```
onOptionsItemSelected() {  
    switch (item.getItemId()) {  
        case R.id.action_settings:  
            startActivity(  
                new Intent(this, SettingsActivity.class));  
            return true;  
        default:  
            return super.onOptionsItemSelected();  
    }  
}
```



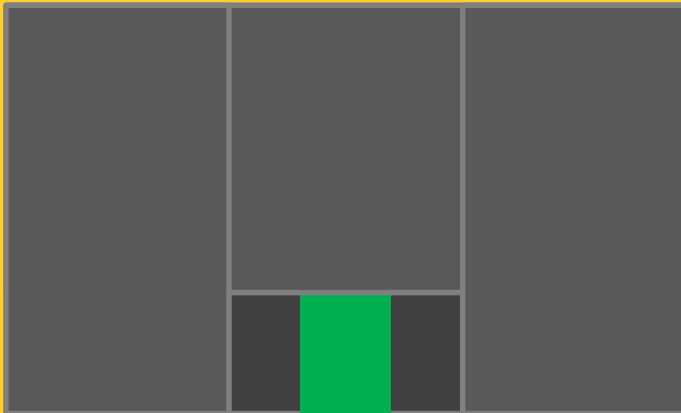
if $i == 1$ (좌측)



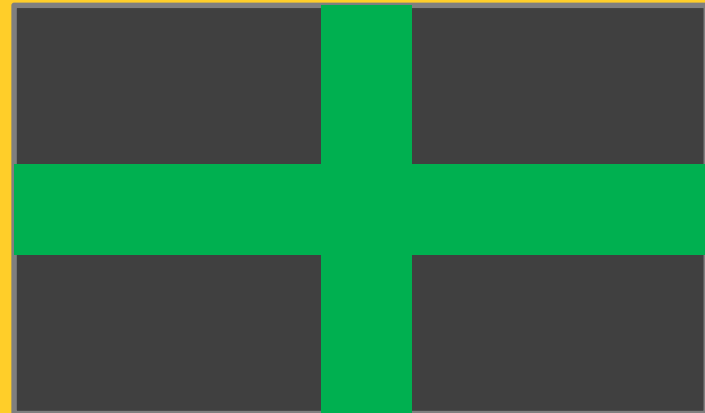
if $i == 2$ (우측)



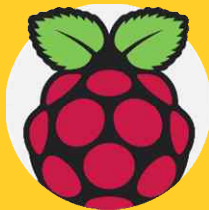
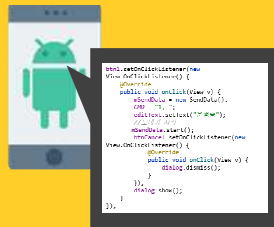
else: // $i = 0$ (중앙)



실제 화면



Line Tracking

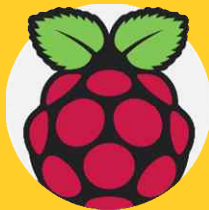
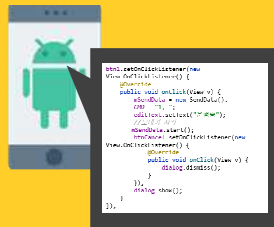


```
Color_Lower = (70,100,70)
Color_Upper = (92, 255, 255)

Frame_Width = 320
Frame_Height = 240

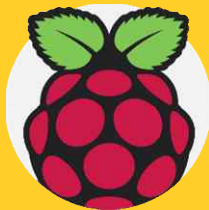
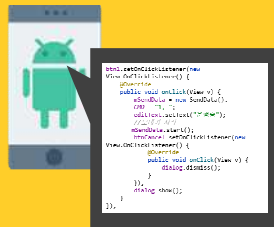
frame_list=[]# [center, left, right]
center = None
center_list=[None, None, None]
(_, frame) = camera.read()
for i in range(3):
    frame_list.append(frame)
    frame_list[i] = cv2.GaussianBlur(frame_list[i], (11, 11),1)
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
    mask = cv2.inRange(hsv, Color_Lower, Color_Upper)
    if i==1:
        cv2.rectangle(mask, (40,0), (320,240), (0,0,0),-1)
    elif i==2:
        cv2.rectangle(mask, (0,0), (280,240), (0,0,0),-1)
    else:
        cv2.rectangle(mask, (0,0), (320,120), (0,0,0),-1)
        cv2.rectangle(mask, (0,0), (120,240), (0,0,0),-1)
        cv2.rectangle(mask, (240,0), (320,240), (0,0,0),-1)
```

Line Tracking



```
frame_list=[]# [center, left, right]
center = None
center_list=[None, None, None]
(_, frame) = camera.read()
for i in range(3):
    frame_list.append(frame)
    frame_list[i] = cv2.GaussianBlur(frame_list[i], (11, 11),1)
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
    mask = cv2.inRange(hsv, Color_Lower, Color_Upper)
    if i==1:
        cv2.rectangle(mask, (40,0), (320,240), (0,0,0),-1)
    elif i==2:
        cv2.rectangle(mask, (0,0), (280,240), (0,0,0),-1)
    else:
        cv2.rectangle(mask, (0,0), (320,120), (0,0,0),-1)
        cv2.rectangle(mask, (0,0), (120,240), (0,0,0),-1)
        cv2.rectangle(mask, (240,0), (320,240), (0,0,0),-1)
    contours, = cv2.findContours(mask, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
```


Line Tracking



```
draw(contours, center_list, frame)
```

```
def draw(contours, center_list, frame):
    if len(contours) > 0:

        c = max(contours, key=cv2.contourArea)

        ((x, y), radius) = cv2.minEnclosingCircle(c)

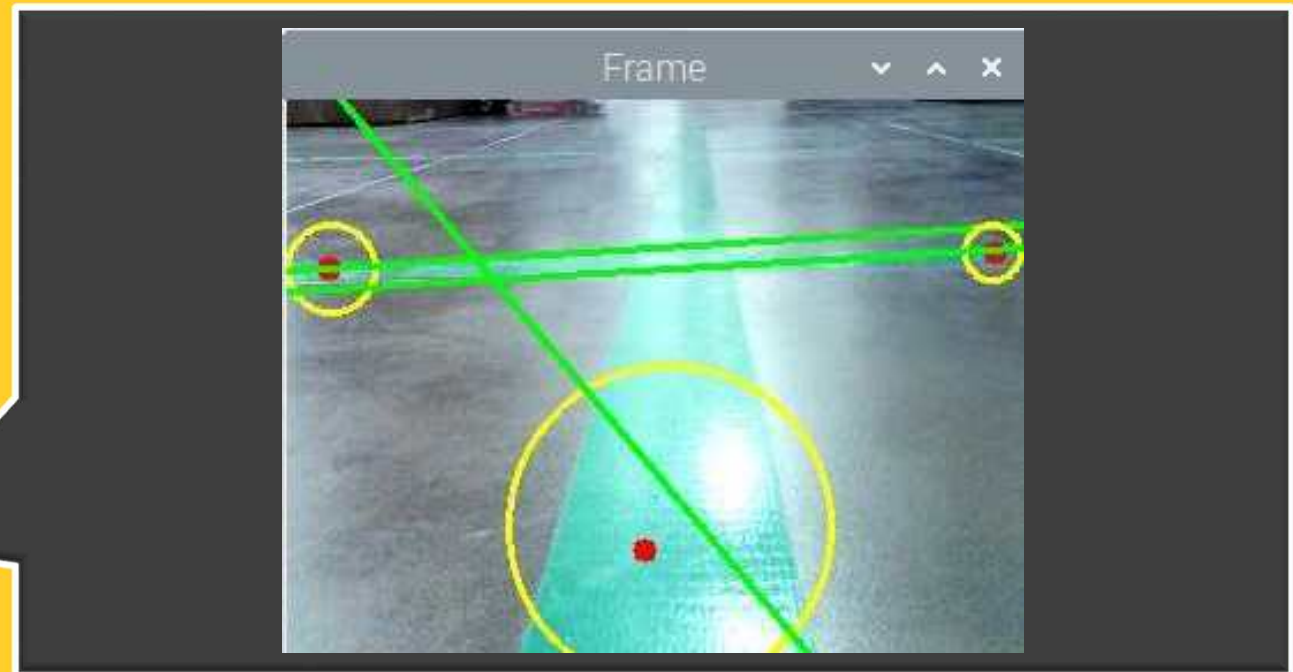
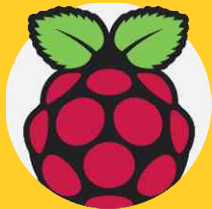
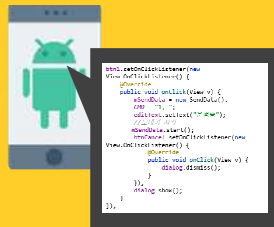
        M = cv2.moments(c)
        try:

            center = (int(M["m10"] / M["m00"]), int(M["m01"] / M["m00"]))
            center_list[i] = center

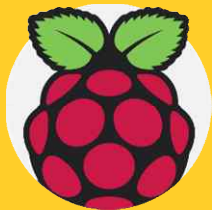
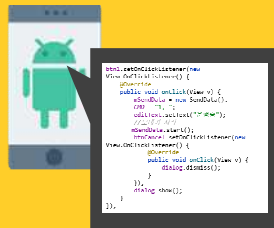
            cv2.circle(frame, (int(x), int(y)), int(radius), (0, 255, 255), 2)
            cv2.circle(frame, center, 5, (0, 0, 255), -1)

            rows, cols = mask.shape[:2]
            [vx, vy, x,y] = cv2.fitLine(c, cv2.DIST_L2, 0, 0.01, 0.01)
            lefty = int((-x*vy/vx)+y)
            righty = int(((cols-x)*vy/vx)+y)
            cv2.line(frame, (cols-1,righty), (0, lefty), (0, 255,0),2)
        except:
            pass
```

Line Tracking



Line Tracking

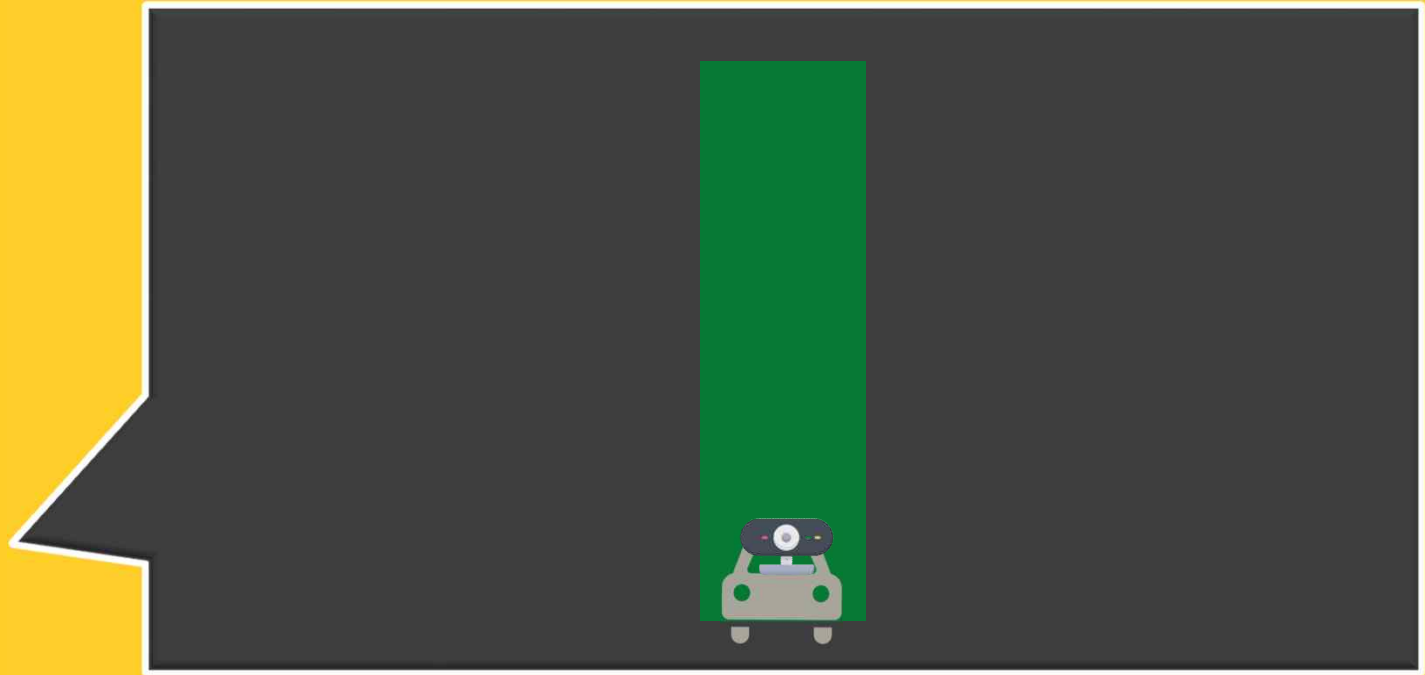
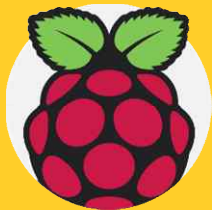


| center_list [0] | center_list [1] | center_list [2] |
|------------------|------------------|------------------|
| 중앙 | 좌측 | 우측 |
| (x,y) | (x,y) | (x,y) |

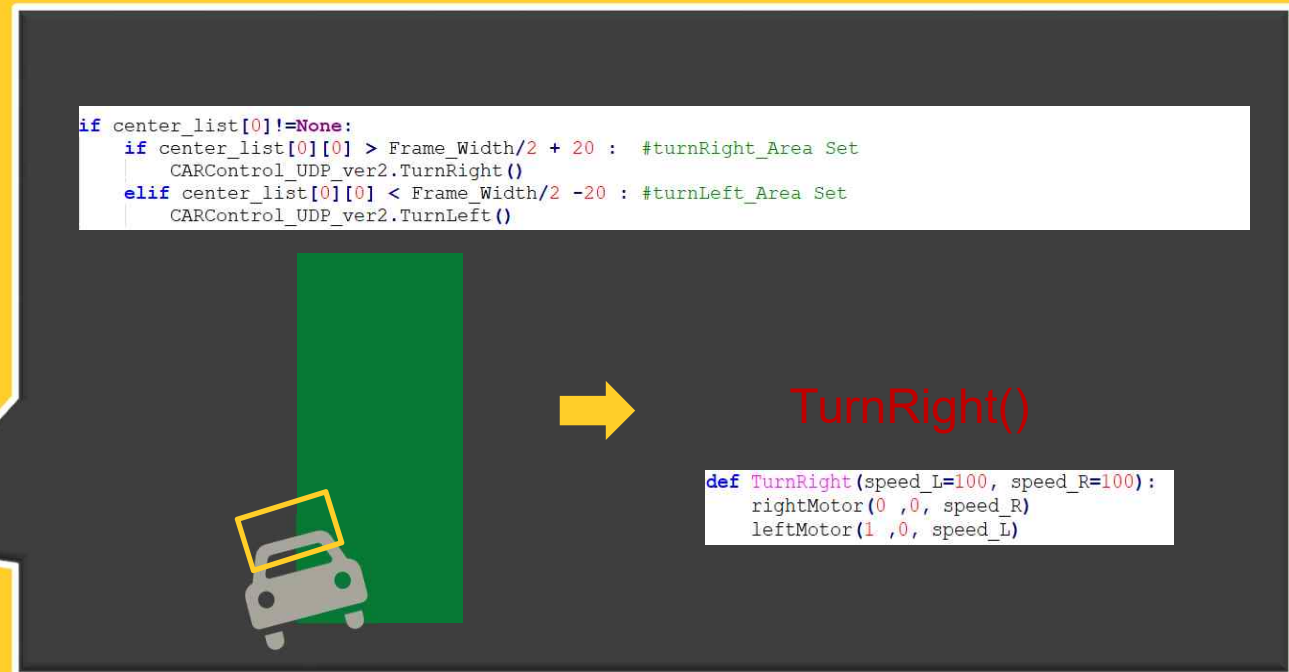
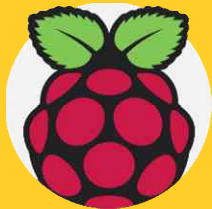
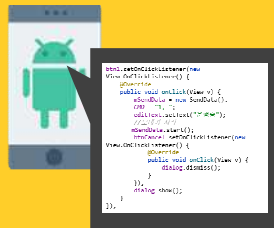
Line Tracking



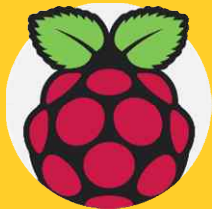
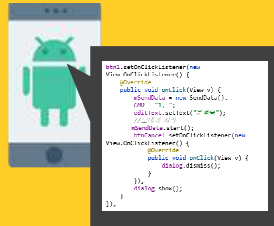
```
onOptionsItemSelected() {  
    switch (item.getItemId()) {  
        case R.id.action_settings:  
            startActivity(  
                new Intent(this, SettingsActivity.class)  
            );  
            return;  
        case R.id.action_logout:  
            startActivity(  
                new Intent(this, LoginActivity.class)  
            );  
            return;  
        default:  
            return;  
    }  
}
```



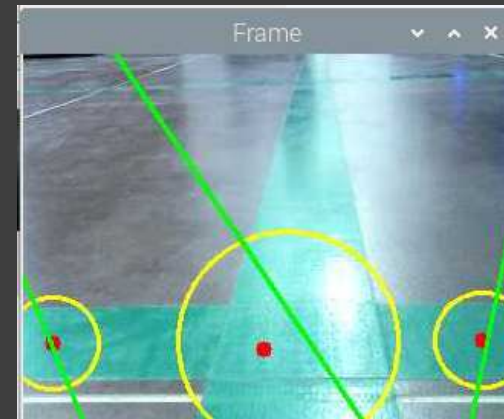
Line Tracking



Line Tracking



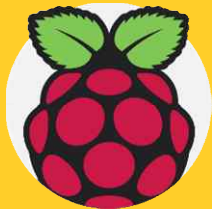
```
if center_list[1] != None and center_list[2] != None :
    ... if center_list[1][1] == Frame_Height/2 :
```



Line Tracking

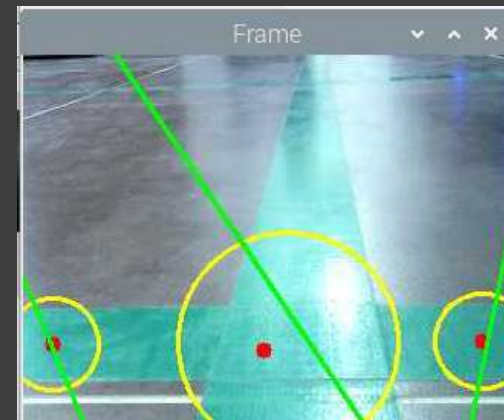


```
1) public void onCreate(Bundle savedInstanceState) {  
2)     super.onCreate(savedInstanceState);  
3)     setContentView(R.layout.activity_main);  
4)     // Initialize variables  
5)     // ...  
6)     // Start the tracking process  
7)     // ...  
8) }
```

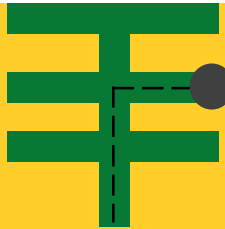


```
crossroad=0  
flag=0
```

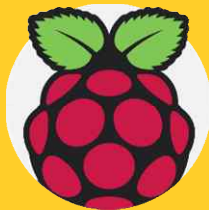
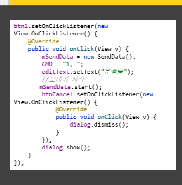
```
else:  
    if center_list[1]!=None and center_list[2]!=None :  
        if center_list[1][1] > Frame_Height/2 and center_list[2][1] > Frame_Height/2 :
```



Line Tracking



destination = 4

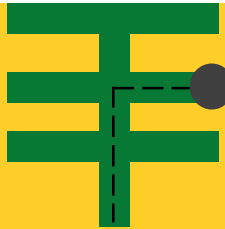


```
crossroad=0
flag=0

else:
    if center_list[1]!=None and center_list[2]!=None :
        if center_list[1][1] > Frame_Height/2 and center_list[2][1] > Frame_Height/2 :
            if flag==0:
                flag=1
                crossroad+=1
            else:
                if destination<3 and crossroad==1:
                    odd_even(int(cmd[0]))
                elif destination<5 and crossroad==2:
                    odd_even(int(cmd[0]))
                else:
                    if crossroad==3:
                        odd_even(int(cmd[0]))
            else:
                flag=0
                CARControl_UDP_ver2.Forward()
                print(crossroad)
                print(flag)
        else:
            CARControl_UDP_ver2.Stop()
```

| crossroad | flag |
|-----------|------|
| 0 | 0 |

Line Tracking



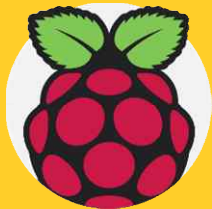
destination = 4



```
import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.View;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.Button;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    /**
     * Called when the activity is first created.
     */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
        fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Snackbar.make(view, "Replace with your own action",
                    Snackbar.LENGTH_LONG).show();
            }
        });
    }
}
```

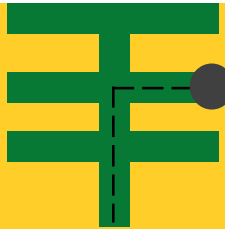


```
crossroad=0
flag=0

else:
    if center_list[1]!=None and center_list[2]!=None :
        if center_list[1][1] > Frame_Height/2 and center_list[2][1] > Frame_Height/2 :
            if flag==0:
                flag=1
                crossroad+=1
            else:
                if destination<3 and crossroad==1:
                    odd_even(int(cmd[0]))
                elif destination<5 and crossroad==2:
                    odd_even(int(cmd[0]))
                else:
                    if crossroad==3:
                        odd_even(int(cmd[0]))
            else:
                flag=0
                CARControl_UDP_ver2.Forward()
                print(crossroad)
                print(flag)
        else:
            CARControl_UDP_ver2.Stop()
```

| crossroad | flag |
|-----------|------|
| 1 | 1 |

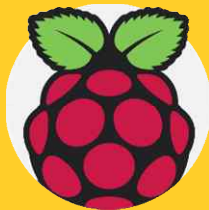
Line Tracking



destination = 4



```
onTouchListener {
    View.OnTouchListener {
        public void onTouch(View v) {
            val id = v.id
            if (id == R.id.button1) {
                // Button 1 pressed
                // ... (code for button 1) ...
            } else if (id == R.id.button2) {
                // Button 2 pressed
                // ... (code for button 2) ...
            }
        }
    }
}
```

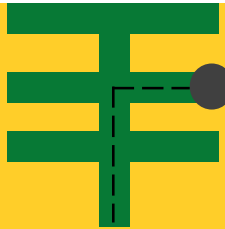


```
crossroad=0
flag=0

else:
    if center_list[1]!=None and center_list[2]!=None :
        if center_list[1][1] > Frame_Height/2 and center_list[2][1] > Frame_Height/2 :
            if flag==0:
                flag=1
                crossroad+=1
            else:
                if destination<3 and crossroad==1:
                    odd_even(int(cmd[0]))
                elif destination<5 and crossroad==2:
                    odd_even(int(cmd[0]))
                else:
                    if crossroad==3:
                        odd_even(int(cmd[0]))
            else:
                flag=0
                CARControl_UDP_ver2.Forward()
                print(crossroad)
                print(flag)
        else:
            CARControl_UDP_ver2.Stop()
```

| crossroad | flag |
|-----------|------|
| 1 | 0 |

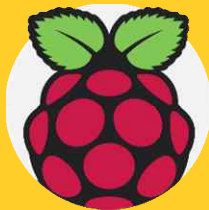
Line Tracking



destination = 4



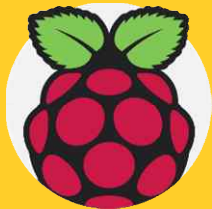
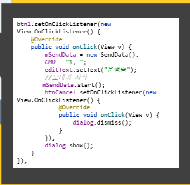
```
onTouchListener {
    View.OnTouchListener {
        public void onTouch(View v) {
            val id = v.id
            if (id == R.id.button1) {
                // Button 1 pressed
                // ... (code for button 1) ...
            } else if (id == R.id.button2) {
                // Button 2 pressed
                // ... (code for button 2) ...
            }
        }
    }
}
```



```
crossroad=0
flag=0

else:
    if center_list[1]!=None and center_list[2]!=None :
        if center_list[1][1] > Frame_Height/2 and center_list[2][1] > Frame_Height/2 :
            if flag==0:
                flag=1
                crossroad+=1
            else:
                if destination<3 and crossroad==1:
                    odd_even(int(cmd[0]))
                elif destination<5 and crossroad==2:
                    odd_even(int(cmd[0]))
                else:
                    if crossroad==3:
                        odd_even(int(cmd[0]))
            else:
                flag=0
                CARControl_UDP_ver2.Forward()
                print(crossroad)
                print(flag)
        else:
            CARControl_UDP_ver2.Stop()
```

| crossroad | flag |
|-----------|------|
| 2 | 1 |



```

else:
    if center_list[1]!=None and center_list[2]!=None :
        if center_list[1][1] > Frame_Height/2 and center_list[2][1] > Frame_Height/2 :
            if flag==0:
                flag=1
                crossroad+=1
            else:
                if destination<3 and crossroad==1:
                    odd_even(int(cmd[0]))
                elif destination<5 and crossroad==2:
                    odd_even(int(cmd[0]))
                else:
                    if crossroad==3:
                        odd_even(int(cmd[0]))
            else:
                flag=0
                CARControl_UDP_ver2.Forward()
                print(crossroad)
                print(flag)
e:
CARControl_UDP_ver2.Stop()

```

| crossroad | flag |
|-----------|------|
| 2 | 1 |

```

def odd_even(destination):
    if (destination%2)==1:
        CARControl_UDP_ver2.TurnLeft()
    else:
        CARControl_UDP_ver2.TurnRight()

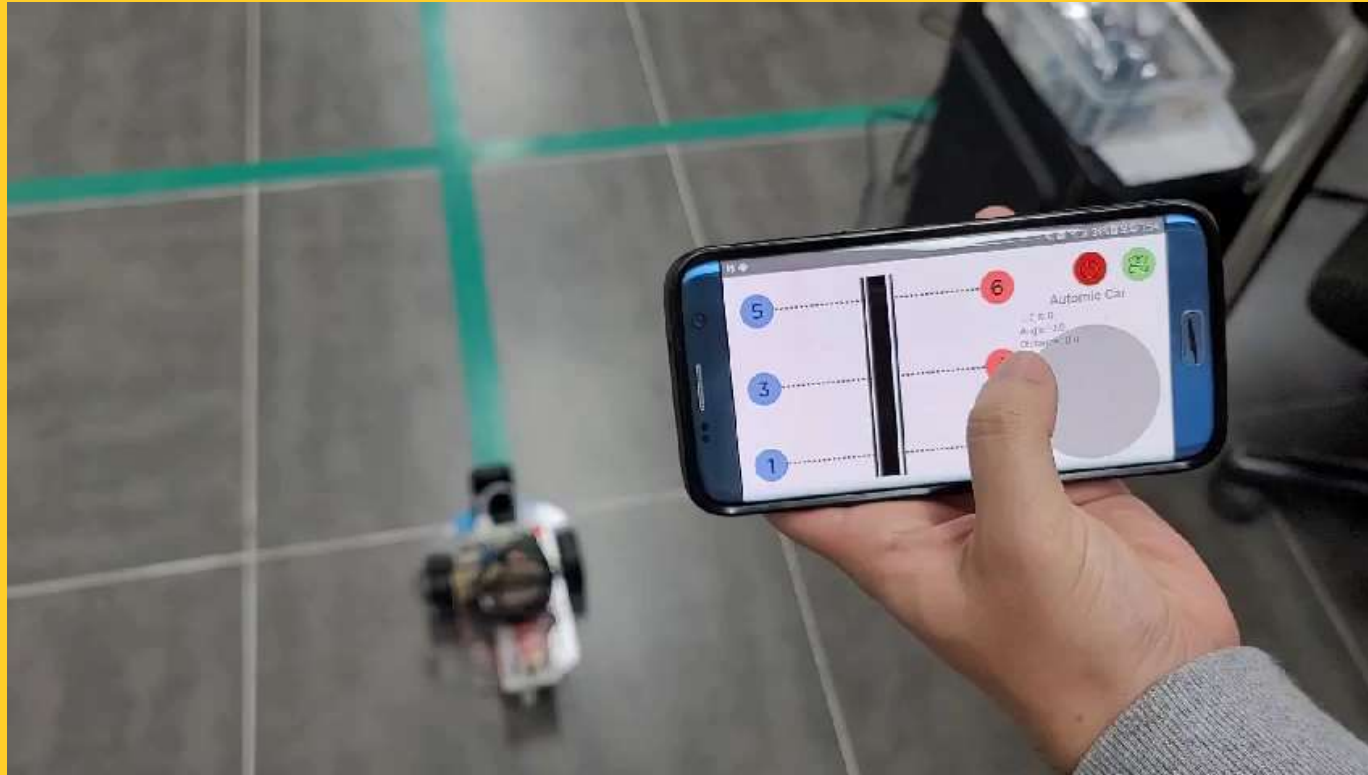
```

| | |
|-----------|------|
| crossroad | flag |
| 2 | 1 |

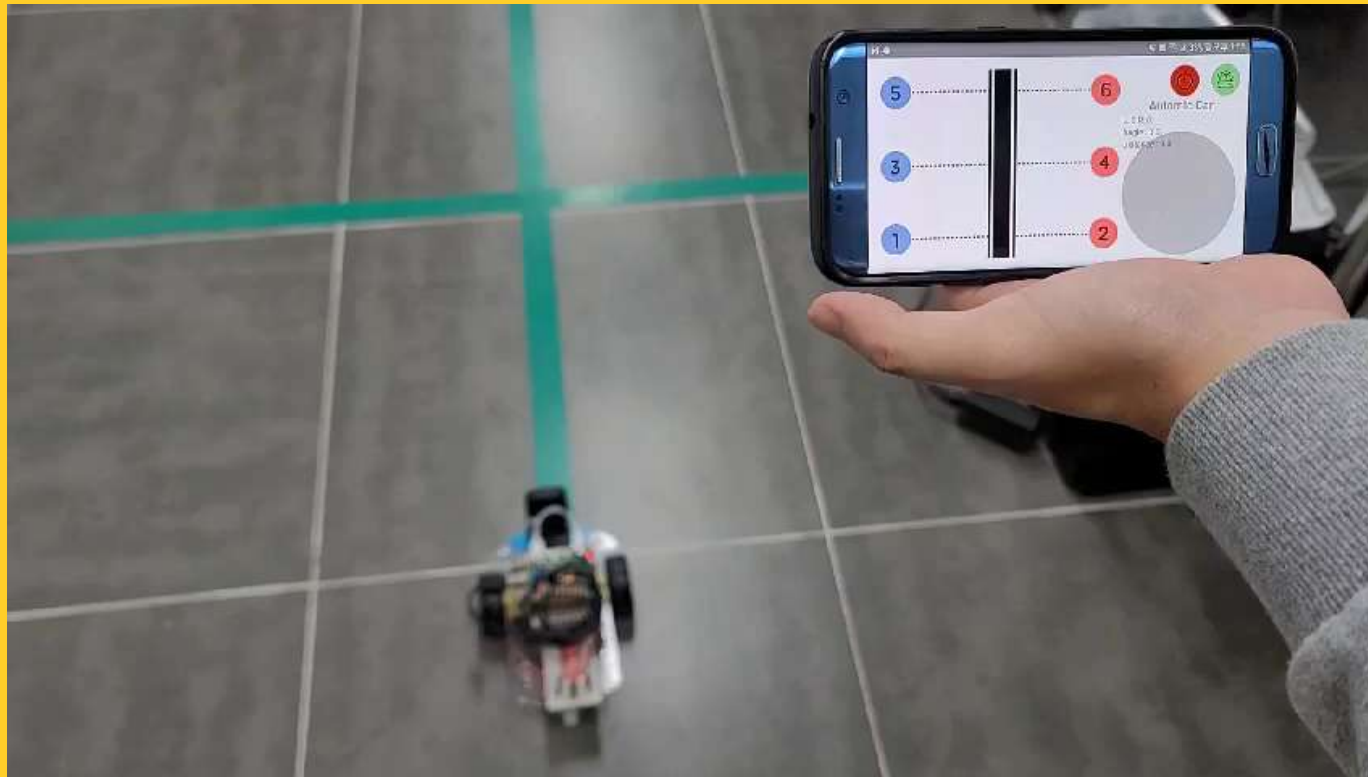
촬영 영상(1번)



촬영 영상(4 번)



촬영 영상(5번)





Thank you