**Autonomous : LineTracer**

**라인트레이서 설계**

창의적공학설계\_8조

# 과제 설명

ROBOTC를 사용하여 EV3 로봇을 제어하여 라인트레이서를 설계합니다. 주어진 트랙 에서 한 개의 색깔 센서를 사용하여 로봇이 특정 색깔 경계를 감지하고 이에 따라 모터를 제어하여 도착점에 도달하는 것을 목표로 합니다.

# 코드 내용

#pragma config(Sensor, S2, c2, sensorEV3\_Color)

#pragma config(Motor , motorA, lm, tmotorEV3\_Large, PIDControl, driveLeft , encoder)

#pragma config(Motor , motorD, rm, tmotorEV3\_Large, PIDControl, driveRight, encoder)

//\*!!Code automatically generated by 'ROBOTC' configuration wizard !!\*//

int calcBound() {

int bound; // 경계값 변수

int black = 0; // 검은색 초기값 설정

int white = 0; // 흰색 초기값 설정

//색깔 센서를 검은색 라인으로 위치

//구동 초기 검은색 반사광 값 인식

for (int i=0; i<5; i++) {

black += getColorReflected(c2);

sleep(10);

}

//색깔 센서를 흰색 라인으로 위치시키기 위해 로봇 회전

//좌회전

setMotorSpeed(lm,0);

setMotorSpeed(rm,40);

sleep(300);

//정지

setMotorSpeed(lm,0);

setMotorSpeed(rm,0);

//구동 초기 흰색 반사광 값 인식

for (int i=0; i<5; i++){

white += getColorReflected(c2);

sleep(10);

}

//검은색과 흰색의 평균으로 경계값 설정

bound = (black/5+white/5)/2;

return bound;

}

task main()

{

int gray = calcBound();

// 무한 루프에서 색깔 센서로 측정한 반사광 값과 경계값을 비교하여, 로봇의 모터를 제어합니다.

while(1) {

if (gray > getColorReflected(c2))

{ /\*센서로 감지한 값이 경계값보다 크면, 로봇은 검은색 영역 위에 있으므로 왼쪽 모터를

강하게 돌려서 오른쪽으로 향하도록 합니다.\*/

setMotorSpeed(lm,50);

setMotorSpeed(rm,10);

}

else

{ /\*로봇은 흰색 영역 위에 있으므로 오른쪽 모터를 강하게 돌려서 왼쪽으로 향하도록 합니다.\*/

setMotorSpeed(lm,10);

setMotorSpeed(rm,50);

}

}

}

# 과제 의의

// 코드의 정확성은 주어진 로봇의 하드웨어 구성 및 센서의 캘리브레이션에 의존하기 때문에

// 특정 환경에서 테스트해 보는 것이 중요합니다.

모든 팀이 같은 부품의 로봇으로, 같은 트랙을 주행합니다. 단, 색깔 센서의 위치는 변경이 가능하기에 로봇의 전후면, 좌우 배치를 결정할 수 있습니다.

저희 팀은 빠른 주행을 목표로 한 과제로서 후진이 불필요하기 때문에 센서를 전면에 배치, 좌우 모터와 회전반경의 대칭성을 고려해 로봇의 중앙에 설치하는 것으로 결정하였습니다.

경계값은 초기 로봇의 동작으로 검은색 반사광값과 흰색 반사광값을 측정하여 평균을 계산하여 설정하였습니다. 보다 정확한 측정을 위해 센서와 바닥과의 높이를 적절하게 설정하였고, 오차 방지를 위해 센서를 튼튼하게 고정하였습니다.

빠른 주행을 위해서는 빠른 모터 속도도 중요하지만 로봇의 라인 이탈을 방지하는 것 또한 중요합니다. 회전 반경이 큰 커브 턴을 사용할 경우 급격한 곡선 구간에서 라인의 변화만큼 회전하지 못해 라인이 이탈될 수 있고, 회전 반경이 작은 회전 방식을 사용하면 라인을 안정적으로 따라갈 수 있지만 속도 감소하기 때문에 경쟁력이 부족할 수 있습니다.

그래서 저희 팀은 모터의 회전 속도를 적절히 제어해야 하도록 반복된 테스트를 통해 회전 속도와 이동 속도를 결정하였습니다.