

스마트센서와 액추에이터 과제중심수업 보고서

ICT융합학부 미디어테크놀로지 전공

2019098068 이찬영

GitHub repository 주소 : <https://github.com/diligentcoder7/Smart-sensor-and-actuator>

(해당 repository에 전체 코드가 올려져 있으니 참고바랍니다.)

1. 코드 리뷰

전처리 구문 & 변수 선언

```
1  #pragma config(Sensor, S1, ts, sensorEV3_Touch)
2  #pragma config(Sensor, S2, gs, sensorEV3_Gyro)
3  #pragma config(Sensor, S3, cs, sensorEV3_Color, modeEV3Color_Color)
4  #pragma config(Sensor, S4, ss, sensorEV3_Ultrasonic)
5  #pragma config(Motor, motorB, lm, tmotorEV3_Large, PIDControl, encoder)
6  #pragma config(Motor, motorC, rm, tmotorEV3_Large, PIDControl, encoder)
7
8  #define Black 1
9  #define Blue 2
10 #define Green 3
11 #define Yellow 4
12 #define Red 5
13 #define White 6
14 #define Brown 7
15
16 int count = 0, start_val = 0, speed = 8;
17 int finish_line, color;
18 int initial, enc_degree;
19 float dist, t;
```

해당 과제 미션을 위해 필요한 전처리 구문들과 제가 임의로 선언한 변수들이 있는 화면입니다.

count는 검은색 선의 개수를 세기 위한 변수, start_val은 시작 지점과 그렇지 않은 지점을 분간하기 위한 변수, speed는 로봇의 주행 속도를 저장하기 위한 변수, finish_line은 출발 지점과 같은 color 값을 저장하기 위한 변수, color는 color sensor로부터 받은 값을 저장하기 위한 변수, initial은 자이로 센서로부터 현재 위치에서 받은 값을 저장하기 위한 변수, enc_degree는 거리를 설정하여 로봇에 맞게 변환한 엔코더 센서 제어값을 저장하기 위한 변수, dist는 초음파 센서로부터 받은 거리 값을 저장하기 위한 변수, t는 타이머를 이용한 시간 값을 저장하기 위한 변수입니다.

※ task main() 내에서 color 값을 define한 번호로 쓸 수도 있었지만 헛갈려서 색깔을 그대로 적었습니다. ex) if(color == Blue)

convert() 함수 & go() 함수

```
21     int convert(float dist)
22     {
23         return (int)(360.0 * dist / 17.58);
24     }
25     |
26     void go(int s)
27     {
28         setMotorSpeed(lm, s);
29         setMotorSpeed(rm, s);
30     }
```

task main() 함수 내에서 쓰이는 함수들입니다.

먼저, convert() 함수는 초음파 센서로부터 장애물과의 거리 값을 인수로 받아서 EV3 바퀴의 둘레인 17.58(cm)을 토대로 엔코더 제어가 얼마만큼 바퀴를 제어해야 할지에 해당하는 값으로 바꾸어 줍니다.

go() 함수는 속도 값을 인수로 받아서 왼쪽 모터와 오른쪽 모터 동시에 같은 속도로 로봇을 전진하게 합니다.

sound() 함수 & point_turn() 함수

```
32     void sound(int n)
33     {
34         playSound(soundBeepBeep);
35         sleep(50);
36         if(n >= 2){
37             for(int i=2; i<=n; i++)
38             {
39                 playSound(soundBeepBeep);
40                 sleep(50);
41             }
42         }
43     }
44
45     void point_turn(int mode, int v)
46     {
47         if(mode == 1){ // point_left_turn
48             setMotorSpeed(lm, -v);
49             setMotorSpeed(rm, v);
50         }
51         else{ // point_right_turn
52             setMotorSpeed(lm, v);
53             setMotorSpeed(rm, -v);
54         }
55     }
```

task main() 함수 내에서 쓰이는 함수들입니다.

sound() 함수는 몇 번 반복할지에 해당하는 값을 인수로 받아 소리를 출력하고, 값이 2 미만일 때는 1번만, 값이 2 이상일 때는 해당 값의 개수만큼 소리를 반복 출력합니다.

point_turn() 함수는 mode 값과 속도 값을 인수로 받아, mode가 1일 때는 해당 속도 값으로 반시계 방향 포인트 턴, mode가 1이 아닐 때는 해당 속도 값으로 시계 방향 포인트 턴 합니다.

left_90() 함수 & right_90() 함수 & black_distance() 함수

```
57 void left_90()
58 {
59     go(0);
60     initial = getGyroDegrees(gs);
61     while(getGyroDegrees(gs) - initial >= -90){ // left point_turn
62         point_turn(1, speed/2);
63     }
64 }
65
66 void right_90()
67 {
68     go(0);
69     initial = getGyroDegrees(gs);
70     while(getGyroDegrees(gs) - initial <= 90){ // left point_turn
71         point_turn(2, speed/2);
72     }
73 }
74
75 void black_distance(int c)
76 {
77     go(0);
78     resetMotorEncoder(lm);
79     enc_degree = convert(c * 10.0);
80     while(getMotorEncoder(lm) < enc_degree)
81     {
82         go(speed/2);
83     }
84 }
```

task main() 함수 내에서 쓰이는 함수들입니다.

left_90() 함수는 EV3 로봇이 현재 위치로부터 반시계 방향으로 90도 만큼, right_90() 함수는 EV3 로봇이 현재 위치로부터 시계 방향으로 90도 만큼 움직이게 합니다.

black_distance() 함수는 검은색 선을 지나간 개수를 인수로 받아 EV3 로봇이 개수 * 10 (cm) 만큼 전진하게 합니다.

2. 진행사항 함수설명

task main()

