

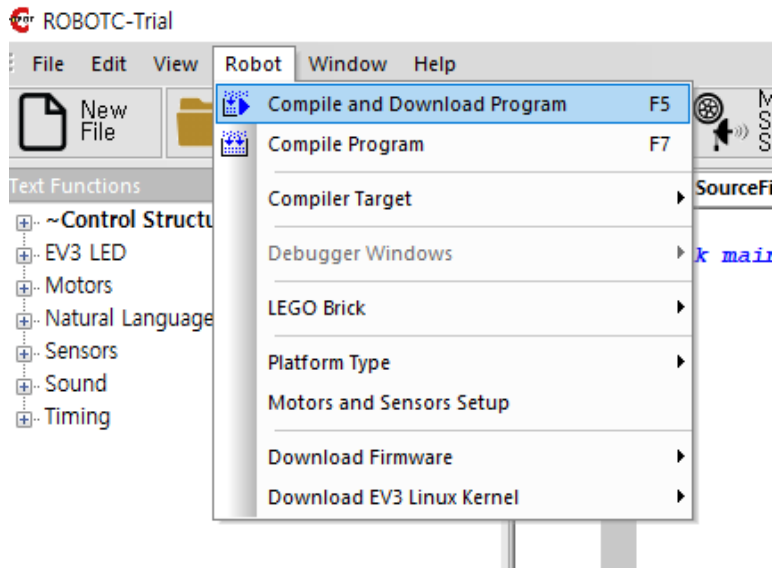
MindStorm EV3

(코드, 모터 및 센서 기초)

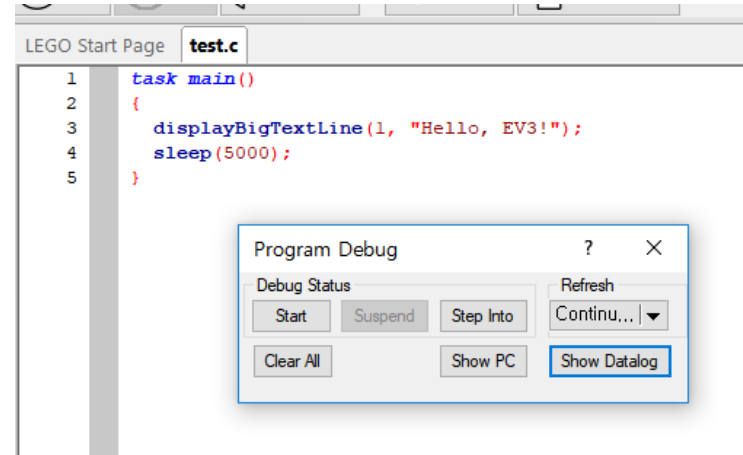
이건희

1. EV3에 코드 심기

줄번호	코드	설명
1	task main()	3: 디스플레이의 1번 줄에 큰 글씨체로 문장을 나타낸다. 4: 현재 상태를 5초간 유지한다.
2	{	
3	displayBigTextLine(1, "Hello, EV3!");	
4	sleep(5000);	
5	}	



- Robot 탭에서 Compile and Download Program을 선택하거나 F5를 눌러서 프로그램을 컴파일 후 다운로드 할 수 있다.



- 다음과 같이 Debug 창이 뜨는데 start 버튼을 눌러 구동을 확인해 볼 수 있다.

1. EV3에서 코드 실행



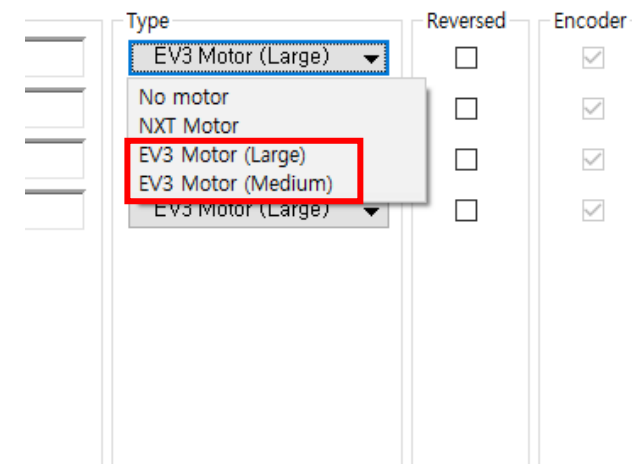
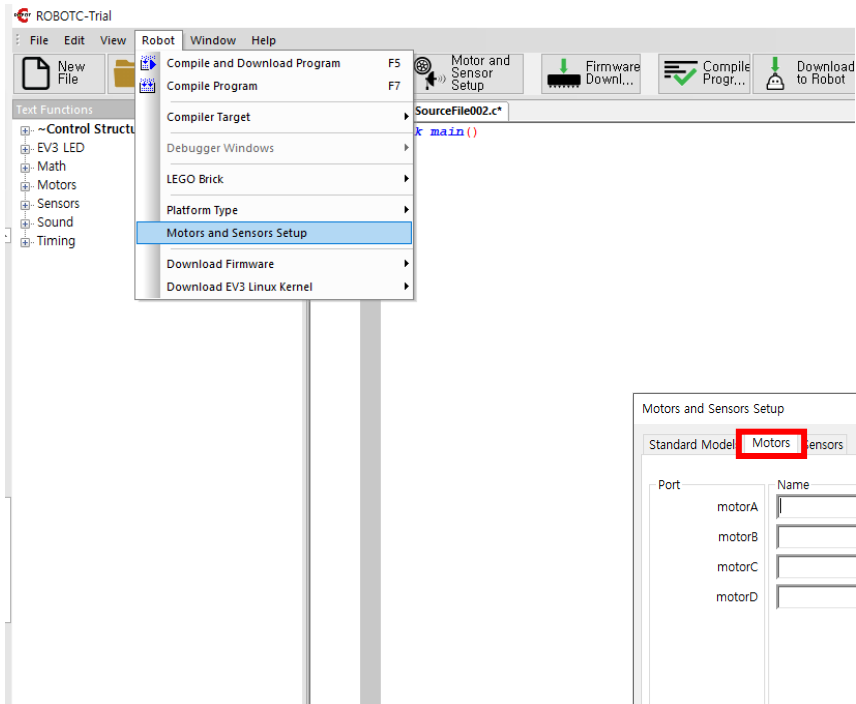
- 최근 실행 탭에서는 가장 최근 실행된 프로그램들을 보여줌



- 파일 내비게이션 탭에서는 SD 카드에 저장된 파일들을 포함하여 EV3 안에 있는 모든 파일들에 접근하고, 파일들의 관리가 가능

2. EV3의 센서와 모터 다뤄 보기

2.1 모터 작동



2. EV3의 센서와 모터 다뤄 보기

2.1 모터 작동



라지 모터



미디엄 모터

줄번호	코드	설명
1	#pragma config(Motor, motorA, rm, tmotorEV3_Large, PIDControl, encoder)	1: 모터 설정, 포트A에 연결(motorA), 이름을 rm이라고 명명
2		EV3 큰 모터, PID제어와 encoder 겸용
3	task main()	5: 모터의 속도 지정, -100에서 100 사이의 정수로 지정
4	{	음수 = 방향이 반대.
5	setMotorSpeed(rm,50);	6: 실행 명령문을 1초동안 유지
6	sleep(1000);	
7	}	

2. EV3의 센서와 모터 다뤄 보기

2.2 함수의 모듈화 예제



라지 모터



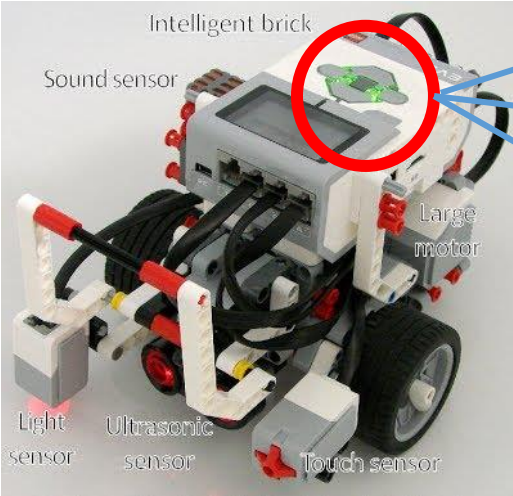
미디엄 모터

줄번호	코드	설명
1	#pragma config(Motor, motorA, rm, tmotorEV3_Large, PIDControl, encoder)	1: 모터 설정, 포트A에 연결(motorA), 이름을 rm이라고 명명 EV3 큰 모터, PID제어와 encoder 검용 6: lm 모터 속도 설정 7: rm 모터 속도 설정 12: 양 모터의 속도를 50으로 지정 (전진) 13: 1초 유지 14: 양 모터의 속도를 50으로 후진 15: 1초 유지
2	#pragma config(Motor, motorD, lm, tmotorEV3_Large, PIDControl, encoder)	
3		
4	void go(int speed_l, int speed_r)	
5	{	
6	setMotorSpeed(lm,speed_l);	
7	setMotorSpeed(rm,speed_r);	
8	}	
9		
10	task main()	
11	{	
12	go(50,50);	
13	sleep(1000);	
14	go(-50,-50);	
15	sleep(1000);	
16	}	

2. EV3의 센서와 모터 다뤄 보기

2.3 LED

음계 \ 옥타브	5	6
도(C)	523	1047
도#(C#)	554	1109
레(D)	587	1175
레#(D#)	622	1245
미(E)	659	1319
파(F)	699	1397
파#(F#)	740	1475
솔(G)	784	1568
솔#(G#)	831	1661
라(A)	880	1760
라#(A#)	932	1865
시(B)	988	1976



- LED Red
- LED Green
- LED Orange

Color Number	동작	Color Number	동작
0	ledOff	5	ledRedFlash
1	ledGreen	6	ledOrangeFlash
2	LedRed	7	ledGreenPulse
3	ledOrange	8	ledRedPulse
4	ledGreenFlash	9	ledOrangePulse

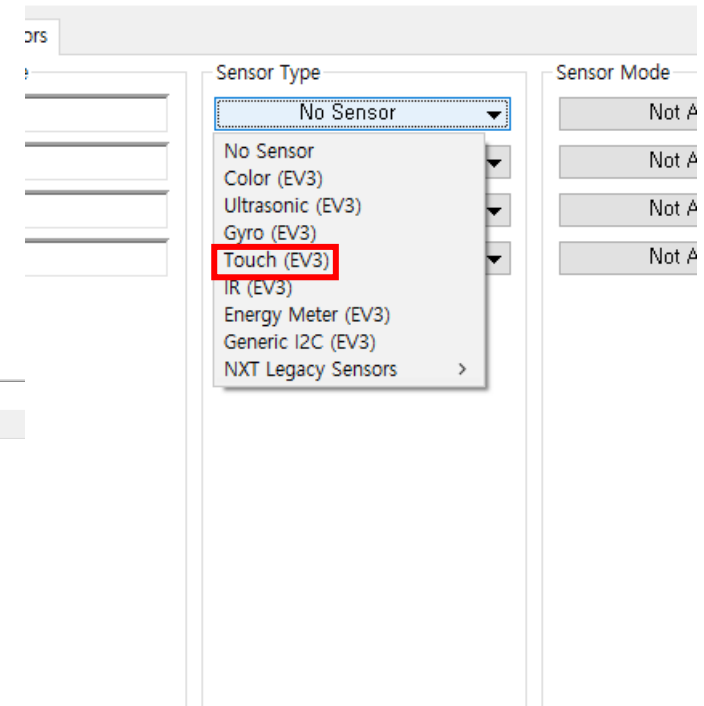
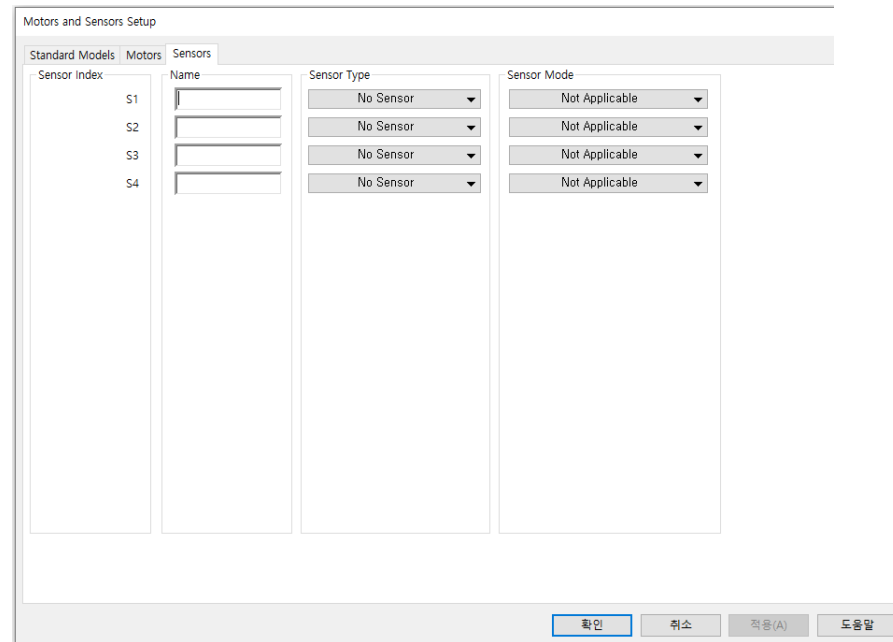
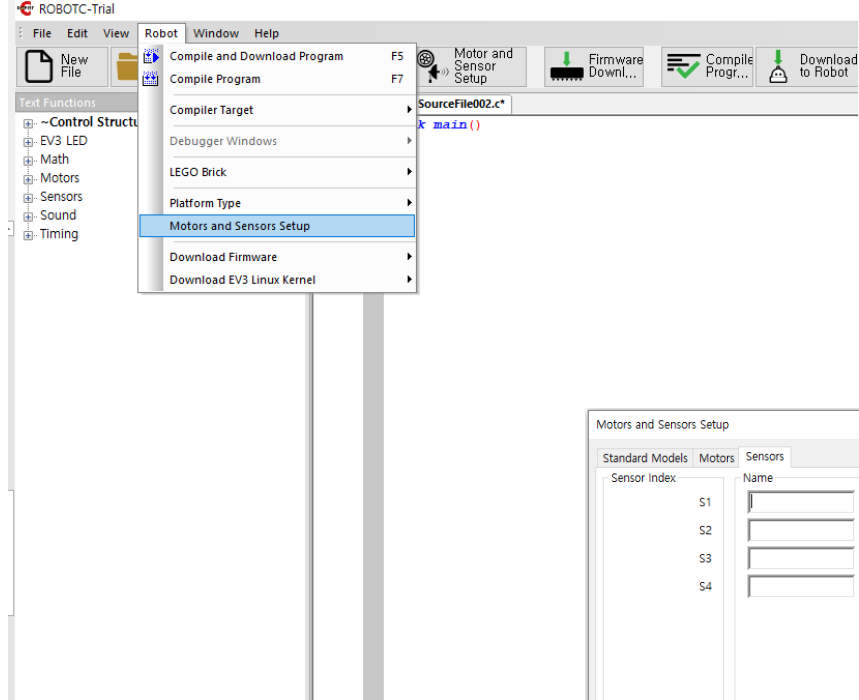
2. EV3의 센서와 모터 다뤄 보기

2.4 Speaker & LED

줄번호	코드	설명
1	task main()	
2	{	
3	int frequency[4] = {523, 587, 659, 784};	3: 크기 4의 배열 선언, 음계는 도,레,미,솔
4	int song[25] = {2,1,0,1,2,2,2,1,1,1,2,3,3,2,1,0,1,2,2,2,1,1,2,1,0}	4: song이라는 배열에 노래를 모두 입력
5	int i, t;	
6	for(i = 0; i<25; i++)	
7	{	
8	t = song[i];	8: 배열의 인덱스를 이용하여 song의 변수를 불러옴
9	playImmediateTone(frequency[t],50);	9: playImmediateTone을 이용하여 각 음계를 불러옴.
10	if(t == 0) setLEDColor(1);	10~13
11	else if(t == 1) setLEDColor(2);	도: LED Green, 레: LED Red 미: LED Orange 솔: LED GreenFlash
12	else if(t == 2) setLEDColor(3);	
13	else if(t == 3) setLEDColor(4);	
14	if(i ==1 i ==14) sleep(250);	14 ~ 17: 박자를 위해 sleep문을 다르게 유지.
15	else if(i == 0 i == 13) sleep(750);	
16	else if(i == 6 i == 9 i == 12 i == 19 i == 24) sleep(1000);	
17	else sleep(500);	
18	}	
	}	

2. EV3의 센서와 모터 다뤄 보기

2.5 터치 센서






2. EV3의 센서와 모터 다뤄 보기

2.5 터치 센서



줄번호	코드	설명
1	#pragma config(Sensor, S1, ts, sensorEV3_Touch)	12: 터치 센서가 눌릴 때까지 아무 것도 하지 않고 현재 상태를 유지한다. 13:터치 센서가 눌렀다가 떼어질 때까지 현재 상태를 유지한다. 14: 현재 속도의 값에 -1을 곱한다. (방향 전환)
2	#pragma config(Motor, motorB, lm, tmotorEv3_Large, PIDControl, encoder)	
3	#pragma config(Motor, motorC, rm, tmotorEv3_Large, PIDControl, encoder)	
4		
5	task main()	
6	{	
7	int speed = 30;	
8	while(1)	
9	{	
10	setMotorSpeed(lm, speed);	
11	setMotorSpeed(rm, speed);	
12	while(getTouchValue(ts) == 0){ }	
13	while(getTouchValue(ts) == 1){ }	
14	speed *= -1;	
15	}	
16	}	

		
눌리지 않은 경우: 센서 반환 값은 0 →while(getTouchValue(ts) == 0){ }에 머물게 됨	눌린 경우: 센서 반환 값은 1 →while(getTouchValue(ts) == 1){ }에 머물게 됨	눌린 경우: 센서 반환 값은 1 →While 문에서 나와 speed*=-1;이 실행됨

2. EV3의 센서와 모터 다뤄 보기

2.6 컬러 센서

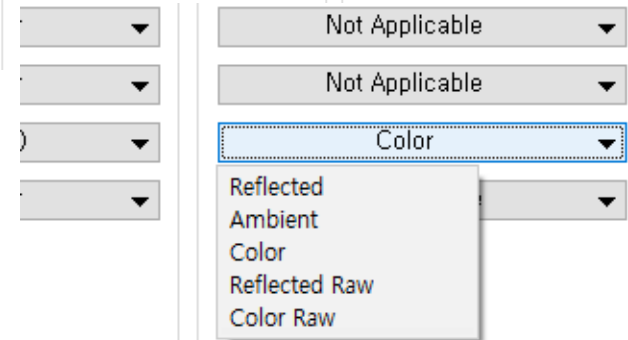
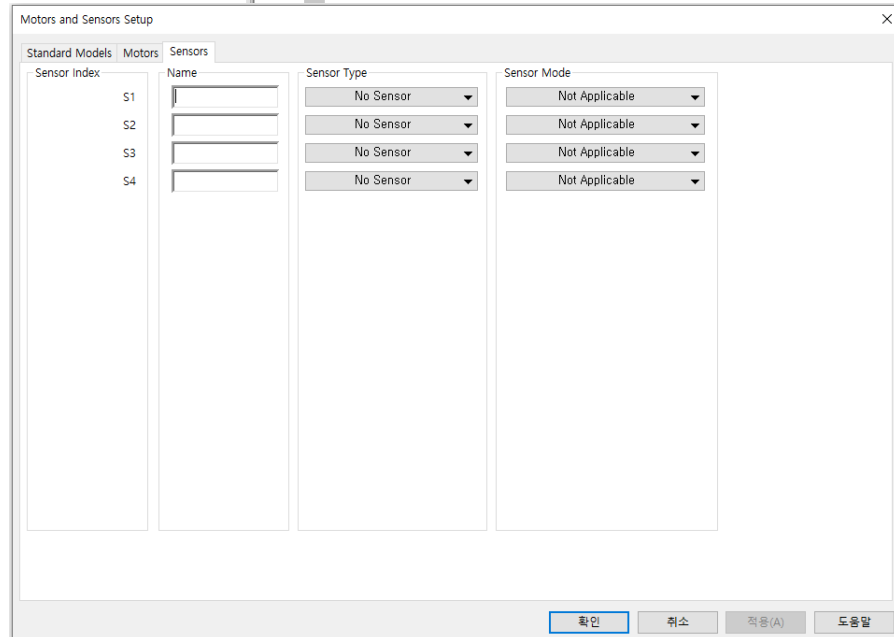
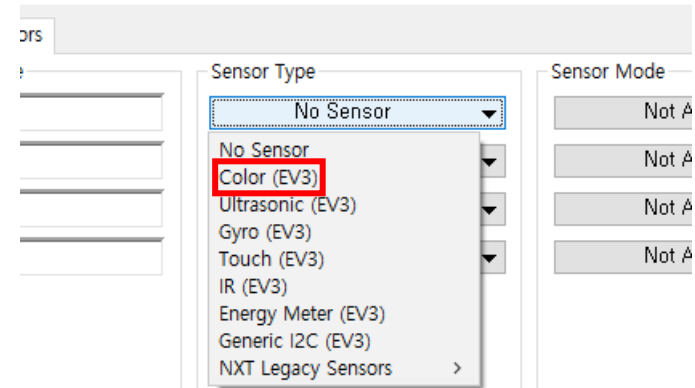
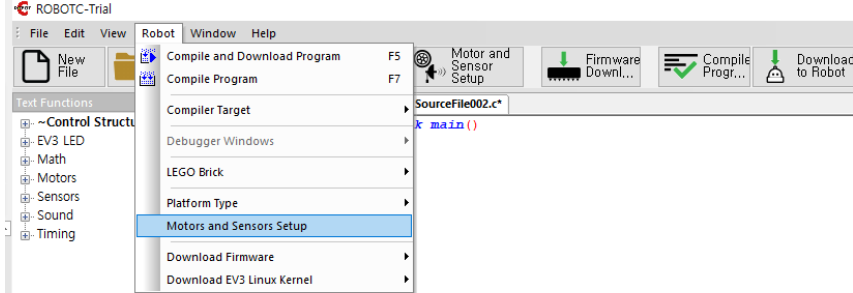


센서 모드	모드 값	설명
Reflected	0(기본값)	반사광 모드, 발광 다이오드에서 발생하는 붉은 빛이 면에 반사되어 들어오는 양을 측정하여 0~100까지의 값으로 반환
Ambient	1	주변광 모드, 주변 빛의 양을 측정하여 0~100사이 값 반환
Color	2	컬러 모드, 0~7 중 색상 값을 반환 (하기 표 참조)

색상 값	색	색상 값	색
0	색상 없음	4	노랑
1	검정	5	빨강
2	파랑	6	흰색
3	초록	7	갈색

2. EV3의 센서와 모터 다뤄 보기

2.7 컬러 센서



2. EV3의 센서와 모터 다뤄 보기

2.7 컬러 센서



줄번호	코드	설명
1	#pragma config(Sensor, S3, cs, sensorEV3_Color)	1: 컬러 센서 설정, 3번 포트 연결, cs라고 명명(reflect mode)
2	task main()	5: 정수형 변수 col 선언
3	{	6: 루프문 시작
4	int col;	8: col 변수에 cs라는 초음파 센서의 값을 받음
5	while(1)	9: EV3 패널 첫 꺾줄에 거리를 표시, %d는 정수형 변수를 넣을 때
6	{	10: 0.5초간 대기
7	col = getColorReflected(cs);	
8		
9	displayBigTextLine(1,"Color: %d", col);	
10	sleep(500);	
11	}	
12	}	

2. EV3의 센서와 모터 다뤄 보기

2.7 컬러 센서



줄번호	코드	설명
1	#pragma config(Sensor, S3, cs, sensorEV3_Color, modeEV3Color_Color)	1: 컬러 센서 설정, 3번 포트 연결, cs라고 명명(color mode)
2	task main()	4: 정수형 변수 col 선언
3	{	5: 루프문 시작
4	int col;	7: col 변수에 cs라는 초음파 센서의 값을 받음
5	while(1)	8~17: switch 문으로 입력 받는 색 출력
6	{	10: 0.5초간 대기
7	col = getColorName(cs);	
8	switch(col){	
9	case 0:displayBigTextLine(1,"Color: No");break;	
10	case 1:displayBigTextLine(1,"Color: black");break;	
11	case 2:displayBigTextLine(1,"Color: blue");break;	
12	case 3:displayBigTextLine(1,"Color: green");break;	
13	case 4:displayBigTextLine(1,"Color: yellow");break;	
14	case 5:displayBigTextLine(1,"Color: red");break;	
15	case 6:displayBigTextLine(1,"Color: white");break;	
16	case 7:displayBigTextLine(1,"Color: brown");break;	
17	}	
18	sleep(500);	
19	}	
20	}	

2. EV3의 센서와 모터 다뤄 보기



2.8 초음파 센서

줄번호	코드	설명
1	#pragma config(Sensor, S1, rss, sensorEV3_Ultrasonic)	1: 초음파 센서 설정, 1번 포트 연결, rss라고 명명
2	task main()	5: 정수형 변수 dist 선언
3	{	6: 루프문 시작
4	int dist;	8: dist 변수에 rss라는 초음파 센서의 값을 받음
5	while(1)	9: EV3 패널 첫째줄에 거리를 표시, %d는 정수형 변수를 넣을 때
6	{	10: 0.5초간 대기
7	dist = getUSDistance(rss);	
8	displayBigTextLine(1,"Distance: %d",dist);	
9	sleep(500);	
10	}	
11	}	
12		

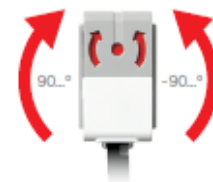
2. EV3의 센서와 모터 다뤄 보기



2.9 초음파 센서

줄번호	코드	설명
1	#pragma config(Sensor, S1, rss, sensorEV3_Ultrasonic)	1: 초음파 센서 설정, 1번 포터 연결, rss라고 명명
2		
3	task main()	
4	{	
5	int dist, pos_x=1;	5: 정수형 변수 dist 선언, x position 선언
6	while(1)	6: 루프문 시작
7	{	
8	if(pos_x>178)	
9	{	
10	eraseDisplay();	10-11: pos_x 가 178을 넘으면 Display를 모두 지움
11	pos_x = 1;	
12	}	
13	dist = getUSDistance(rss);	13: rss값 받음
14	drawLine(pos_x, 0, pos_x,dist);	14: Display에 그래프 표시
15	sleep(200);	15: 0.2초 대기
16	pos_x += 2;	16: pos_x 를 루프 한번 마다 원래 값에 2를 더해 줌
17	}	
18	}	

2. EV3의 센서와 모터 다뤄 보기



2.10 자이로 센서

줄번호	코드	설명
1	#pragma config(Sensor, S1, gs, sensorEV3_Gyro)	1: 자이로 센서 설정, 1번 포트 연결, gs라고 명명
2	task main()	4: 정수형 변수 initial, rot 선언
3	{	5: 자이로 센서 초기값 저장
4	int initial, rot;	6: 루프문 시작
5	initial = getGyroDegrees(gs);	8: rot 변수에 gs라는 초음파 센서의 값을 받고 초기값을 빼줌
6	while(1)	9: EV3 패널 첫째줄에 각도를 표시, %d는 정수형 변수를 넣을 때
7	{	10: 0.5초간 대기
8	rot = getGyroDegrees(gs)- initial;	
9	displayBigTextLine(1,"Degree: %d", rot);	
10	sleep(500);	
11	}	
12	}	