CNRLAB

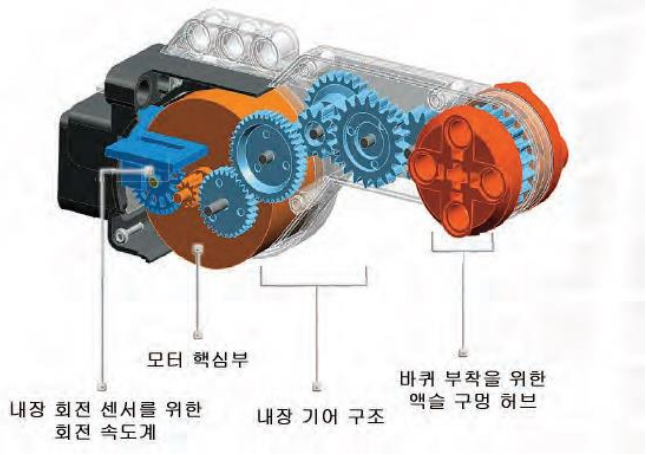
CIM&ROBOTICS LABORATORY

10주차 엔코더를 이용한 엔코더로봇

2015.11.02

기초로봇공학실험

엔코더 (Encoder)



✓ 엔코더

: 물체의 위치 혹은 속도를 출력하는 센서

- 모터의 회전 방향과 각도를 출력.
- 로봇의 모터를 정밀하게 제어하기 위해 주로 사용
- 서보 모터 내부에 엔코더 내장
- 1도 단위로 엔코더 값 출력 (작동오차 : ±1도)

● 엔코더를 활용한 모터 응용

- ➢ 원하는 각 만큼 모터 회전
- ➢ 바퀴를 원하는 회전 수 만큼 회전
- ➢ 계산을 통해 원하는 거리 이동
- ➢ 회전량 체크

엔코더 확인

1. Output 포트 연결

2. View 메뉴

3. Motor Encoder (D / R)

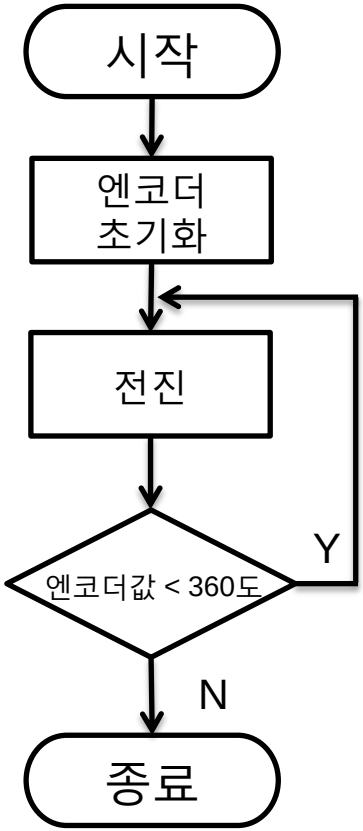
4. Port 선택

5. Result

엔코더 실습

➢ 예제 1 : 원하는 각도만큼 바퀴 회전

✓ 조건 : 한 바퀴(360도) 회전한 후 정지



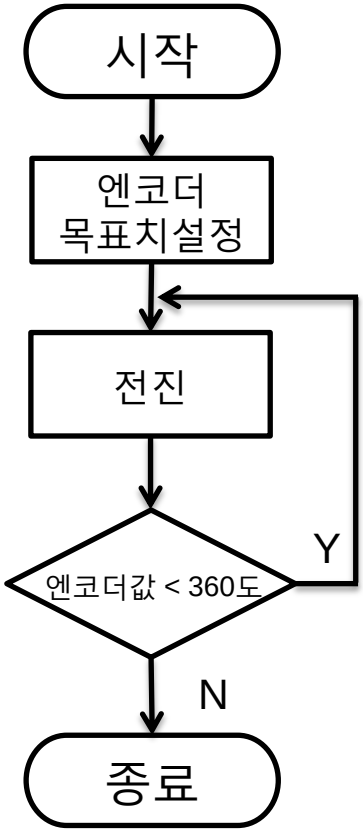
모터의 엔코더 값이 저장되는 내장변수

```
task main()
{
  nMotorEncoder[motorA] = 0;
  do
  {
    motor[motorA] = 40;
    motor[motorC] = 40;
  }
  while(nMotorEncoder[motorA] <360 );
}
```

엔코더 실습

➢ 예제 2 : 원하는 각도만큼 바퀴 회전(2)

✓ 조건 : 한 바퀴(360도) 회전한 후 정지 → 보다 정밀한 제어



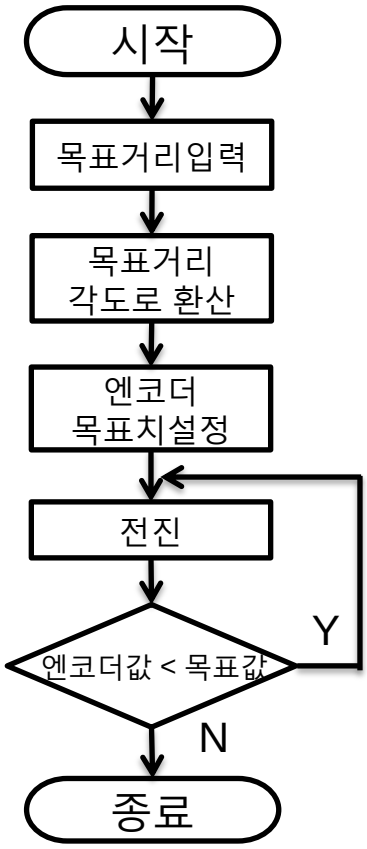
천천히 모터를 정지시키며
목표 값을 지나면 전원이 들어가지 않는 상태로 바꿈

```
task main()
{
  nMotorEncoderTarget[motorA] = 360;
  nMotorEncoderTarget[motorC] = 360;
  motor[motorA] = 40;
  motor[motorC] = 40;
  while(nMotorEncoder[motorA] < 360);
}
```

엔코더 실습

➢ 예제 3 : 원하는 거리만큼 전진

✓ 조건 : 바퀴지름 – 56mm / 200mm 전진하고 정지시킬 것

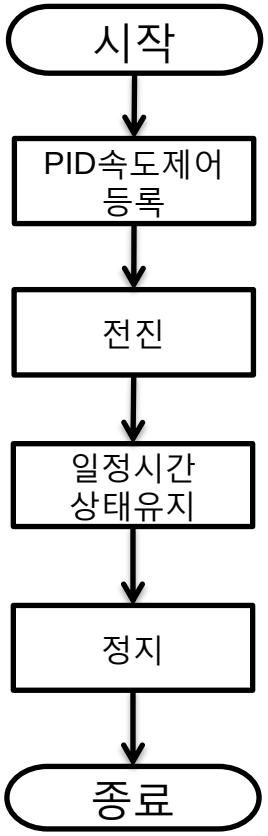



```
#define DIAMETER 56

float convert(float distance)
{
  float angle;
  angle = (360 * distance / DIAMETER / PI);
  return angle;
}

task main()
{
  int rotation;
  int distance = 200;

  rotation = (int) convert((float)distance);

  nMotorEncoderTarget[motorA] = rotation;
  nMotorEncoderTarget[motorC] = rotation;
  motor[motorA] = 40;
  motor[motorC] = 40;

  while(nMotorEncoder[motorA] < rotation);
}
```

캐스트 연산 : 강제로 자료형 바꿈
< 자동변환 시 왼쪽 자료형에 맞춤 >

엔코더 실습

➢ 예제 4 : PID 제어를 이용한 속도제어



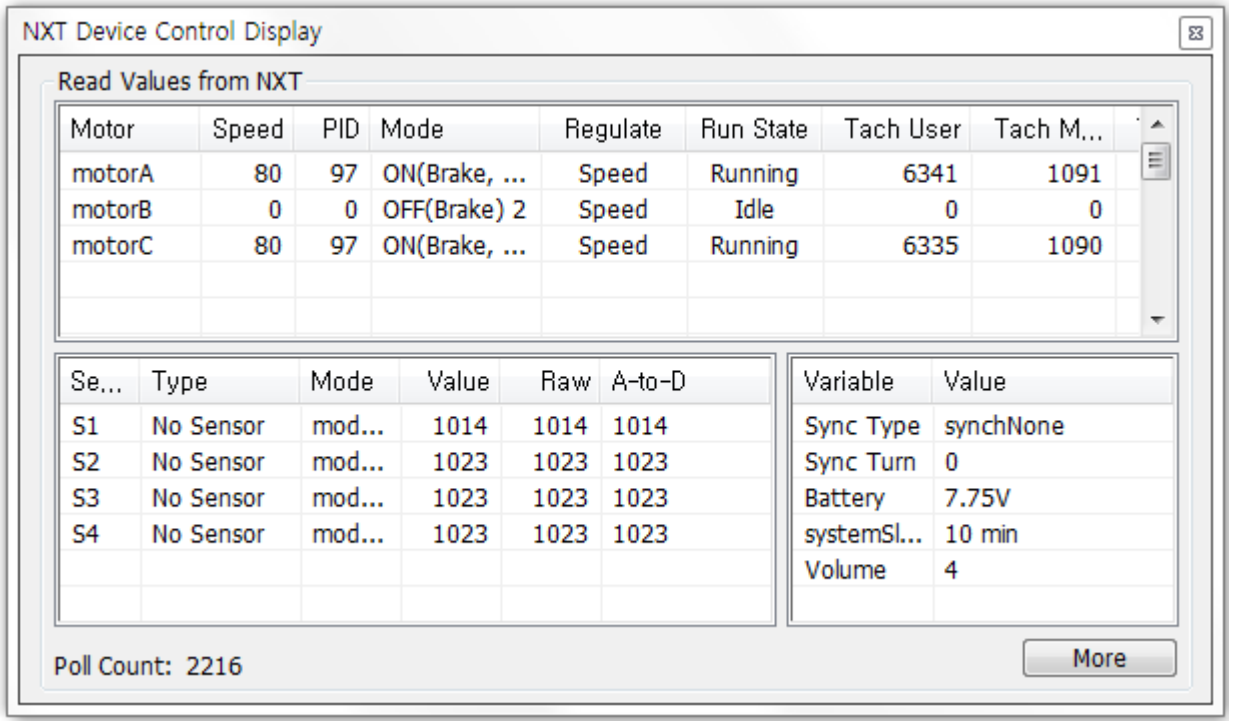
모터 PID 속도제어 설정
- mtrSpeedReg : 속도제어 설정
- mtrNoReg : 속도제어 해제

```
task main()
{
  nMotorPIDSpeedCtrl[motorA] = mtrSpeedReg;
  nMotorPIDSpeedCtrl[motorC] = mtrSpeedReg;

  motor[motorA] = 80;
  motor[motorC] = 80;

  wait1Msec(10000);

  motor[motorA] = 0;
  motor[motorC] = 0;
}
```

엔코더 실습 - PID

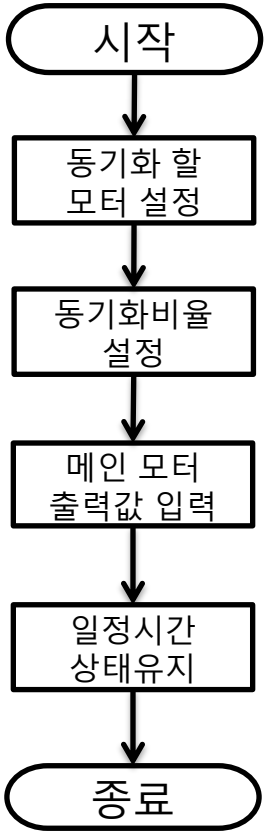
PID 제어

- 모터의 정확한 출력을 위한 제어기법 (오차제어)
- 배터리의 상태에 상관없이 속도출력 동일하게 제어 가능



엔코더 실습

➢ 예제 5 : 모터 동기화 (Synchronizing)




```
task main()
{
  nSyncedMotors = synchAC;
  nSyncedTurnRatio = 100;
  motor[motorA] = 40;
  wait1Msec(4000);
}
```

모터 동기화 설정
- synchAC : A에 C를 동기화
- synchNone : 동기화 해제

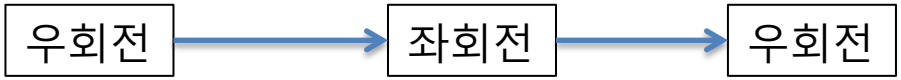
모터 동기화 비율 설정

→ $C_{출력} = \frac{동기화\ 비율}{100} \times A_{출력}$

< 10주차 미션 >

- 엔코더를 이용한 로봇 회전
  - ①포인터턴 / ②스윙턴 / ③커브턴 구현
  - NXT 버튼을 누르면 회전 모드 변경
  - 오른쪽 버튼 클릭 시 : → 왼쪽 버튼은 반대순서로.

  - 엔터 버튼 클릭 시 :

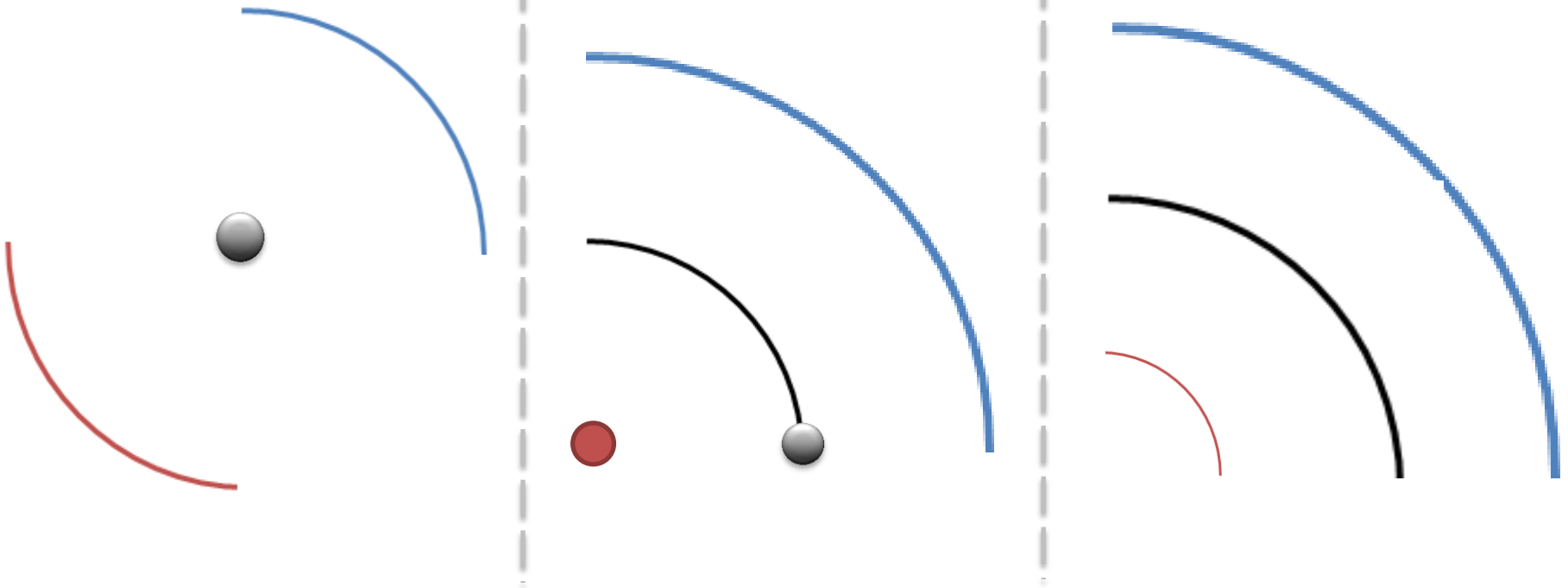


엔코더를 이용한 회전

포인트 턴

스윙 턴

커브 턴



< 10주차 미션 >

➢ Performance

항목	세부 내용	배점
엔코더로봇	세 가지 회전모드 구현	3
	좌우 버튼 기능 구현	3
	엔터 버튼 기능 구현	2
	1시간 이내에 성공	2

➢ Algorithm & Programming

항목	세부 내용	배점
순서도	순서도	6
프로그램 능숙도	소스코드	4

과제(11주차 제출)

다음주 **일요일**까지

hshhln5@gmail.com 에 제출

예비 레포트

✓ 블루투스

- 작동원리 / 센서종류 / 적용분야

결과 레포트

- 로봇 구동 알고리즘 설명
  - ① Source Code
  - ② 순서도
- Discussion
  - ① 기술적 문제점
  - ② 문제 해결 방안