제16회 임베디드 소프트웨어 경진대회
The World Embedded Software Contest 2018

FUNERS

주니어 임베디드SW 챌린저

-2차 기술 지원 교육-

2018. 08. 04

진행 순서

FUNERS

오전	휴식	오후
자이로 센서의 특징		배열
자이로 센서를 이용한 회전 보정		반복문으로 배열 탐색
자이로 센서를 이용한 직진 보정		배열의 최소값, 최댓값, 평균값 찾기
모터 회전각을 이용한 직진 보정		2차 배열
컬러센서를 이용한 위치 보정		디스플레이에 정보 표시하기
		2단계 미션 소개

자이로 센서

자이로 센서

- 자이로 센서
  - 자이로 센서 객체를 생성하고 객체 안의 함수를 이용하여 제어한다.
- 센서 값 제어
  - GyroSensor(입력 포트) – 자이로 센서 객체를 생성
  - mode = 센서 모드 – 센서 모드를 지정,
    - ‘GYRO-ANG’, ‘GYRO-G&A’, ‘GYRO-RATE’
  - value() – 센서가 감지하고 있는 값

자이로 센서

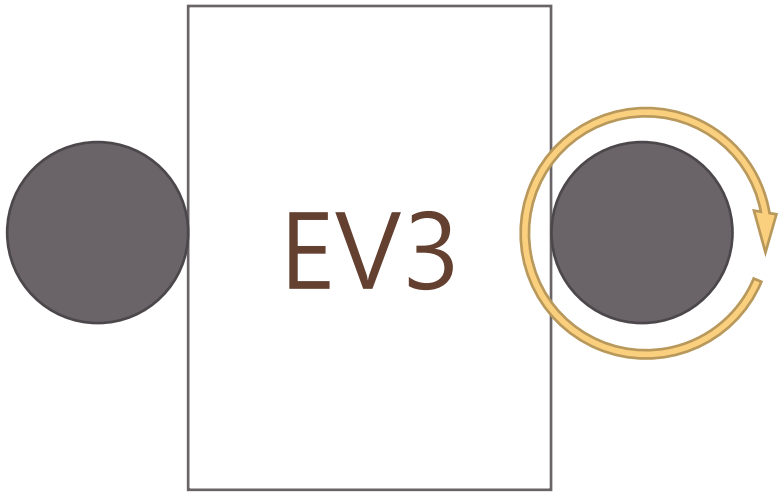
FUNERS

- EV3 자이로 센서의 특징
  - 움직임이 없어도 센서 값이 계속 변화하는 경우가 있다
  - 유일한 해결책은 연결선을 물리적으로 뽑았다가 다시 꽂는 방법 뿐
  - 센서의 방향은 회전의 축과 센서의 위쪽면이 수직이 되도록 설치해야 한다.
  - 시계방향 회전은 +, 반시계방향은 – 로 값이 변화한다.

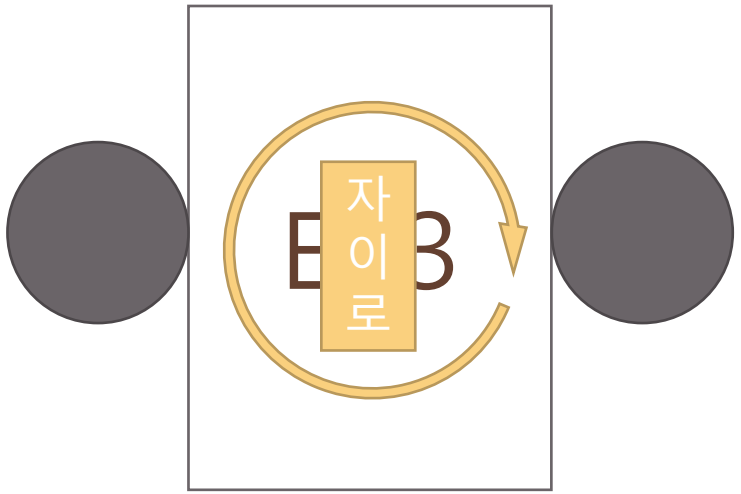
자이로 센서

FUNERS

- 자이로 센서를 이용한 회전
  - 모터의 회전각과 비슷하지만 다르다
  - 처음 초기화 된 방향을 기준으로 0도 표시한다



모터 기준 360도 회전



센서기준 90도 회전

자이로 센서

FUNERS

- 자이로 센서를 이용해 90도 회전 후 제자리

```
motorB = LargeMotor('outB')
gyro = GyroSensor('in4')
gyro.mode = 'GYRO-ANG'

motorB.run_forever(speed_sp=300)

while True:
    if gyro.value() > 90:
        break

motorB.stop(stop_action="hold")

wait(500)

motorB.run_forever(speed_sp= -300)

while True:
    if gyro.value() <= 0:
        break

motorB.stop(stop_action="hold")
```

직진을 해보자

자이로 센서

FUNERS

- 직진이 잘 안되는 이유
  - 모터의 상태 혹은 전류의 차이
  - 로봇의 좌우 대칭 문제
  - 바닥면의 마찰력, 기울어짐 차이
  - 타이어의 마모
  - 기타 등등…

자이로 센서

FUNERS

- **자이로 센서를 이용한 좌우 속도 보정**
  - 현재 바라보고 있는 방향을 기준 (예시: 90도)
  - 직진을 한다면 90도로 유지가 되어야 한다.
  - 틀어진다면 90에서 값이 변한다.
  - 기준이 되는 방향에서 틀어진 만큼 모터 파워를 감속 혹은 가속 시켜준다.

  - 모터B 파워 = 300 – (기준 90 – 현재 방향)
  - 모터C 파워 = 300 + (기준 90 – 현재 방향)

자이로 센서

FUNERS

- 자이로 센서를 이용해 직진 보정

```
motorB = LargeMotor('outB')
motorC = LargeMotor('outC')
gyro = GyroSensor('in4')
gyro.mode = 'GYRO-ANG'
pivot = gyro.value()

motorB.position = 0
motorC.position = 0

while True:
    error = pivot-gyro.value()
    motorB.run_forever(speed_sp=300+error)
    motorC.run_forever(speed_sp=300-error)
    if motorB.position >= 3000:
        break

motorB.stop(stop_action="hold")
motorC.stop(stop_action="hold")
```

자이로 센서

FUNERS

- 모터의 회전각을 이용한 좌우 속도 보정
  - 양쪽 모터의 회전각을 0 으로 초기화 한다.
  - 양쪽 바퀴의 크기가 같을 때 직진을 한다면 두 모터의 회전각도가 같을 것이다.
  - 직진을 하지 못할 경우 양쪽 회전 각도가 다를 것이다.
  - 두 모터의 회전각도가 다른 만큼 모터의 파워를 감속 혹은 가속 시켜준다.

  - motorB.run_forever(speed_sp=300+회전각차이)
  - motorC.run_forever(speed_sp=300-회전각차이)

자이로 센서

FUNERS

- 회전각도 차를 이용해 직진 보정

```
motorB = LargeMotor('outB')
motorC = LargeMotor('outC')

motorB.position = 0
motorC.position = 0

while True:
difference = (motorB.position - motorC.position)*10
motorB.run_forever(speed_sp=300-difference)
motorC.run_forever(speed_sp=300+difference)
if motorB.position >= 3000:
break

motorB.stop(stop_action="hold")
motorC.stop(stop_action="hold")
```

위치 보정

위치 보정

- 위치 보정
  - 로봇의 위치를 항상 정확하게 유지하는 것이 이번 미션의 가장 중요한 핵심 요소이다
  - 자율 주행 자동차, 자동 항법 시스템, 자동 조립 로봇 팔 등
  - GPS, 레이저 측정, 적외선, 카메라, 물리적인 제한 등

위치 보정

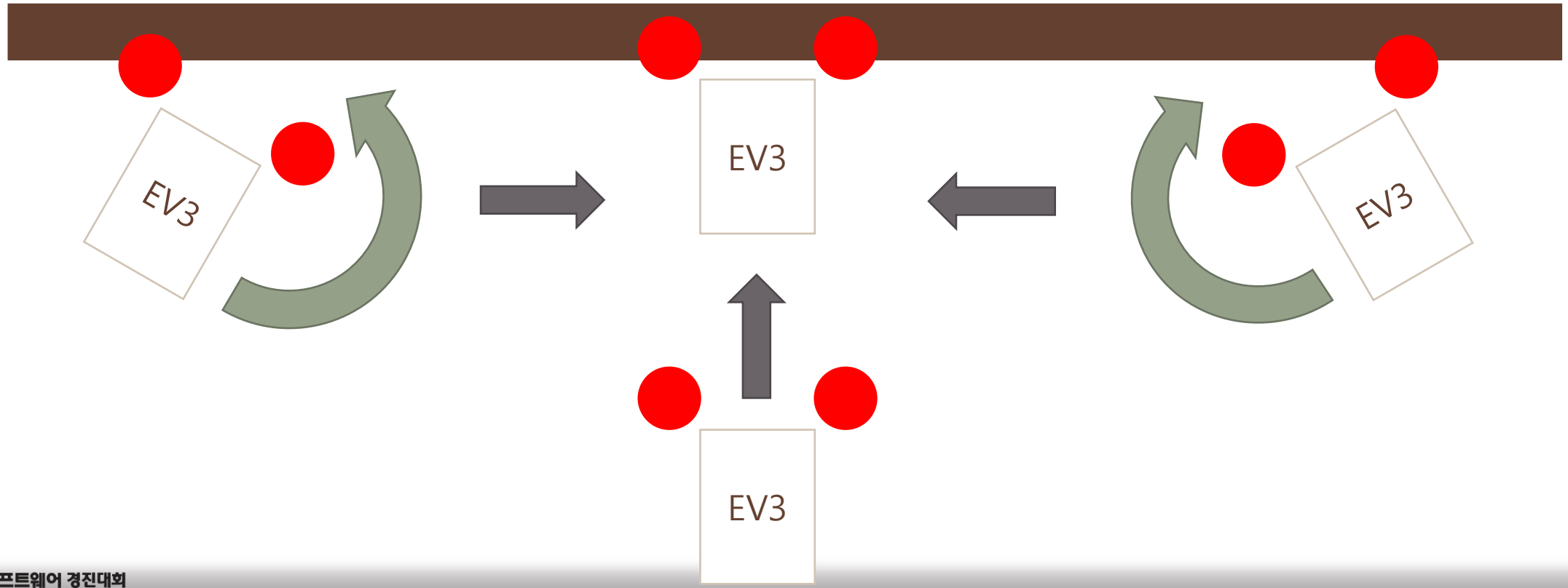
FUNERS

- 컬러센서 2개와 검은 선을 이용한 보정
  - 로봇의 앞쪽에 같은 선상에 부착된 2개의 컬러센서를 이용한다
  - 직선으로 되어있는 검은 선을 기준으로 컬러센서 2개를 검은 선 위로 각각 위치시킨다.
  - => 로봇이 검은 선과 수직하는 방향으로 멈추게 된다.

위치 보정

FUNERS

- 컬러센서 2개와 검은 선을 이용한 보정



위치 보정

FUNERS

- 컬러센서 2개와 검은 선을 이용한 보정

```
while True:
    if color_left.value() < 20 :
        if color_right.value() < 20 :
            motorB.run_forever(speed_sp=0)
            motorC.run_forever(speed_sp=0)
            break
        else :
            motorB.run_forever(speed_sp=0)
            motorC.run_forever(speed_sp=300)
    else :
        if color_right.value() < 20 :
            motorB.run_forever(speed_sp=300)
            motorC.run_forever(speed_sp=0)
        else :
            motorB.run_forever(speed_sp=300)
            motorC.run_forever(speed_sp=300)

motorB.stop(stop_action="hold")
motorC.stop(stop_action="hold")
```

배열

배열

FUNERS

- **배열 – 사물함, 기차, 아파트, 여러 칸이 있는 그릇**
  - 여러 개의 데이터를 하나의 이름과 번호로 저장 할 수 있는 변수
  - 저장된 데이터의 번호는 0부터 시작한다.
  - 배열을 사용하지 않을 경우:
    - 1000개의 데이터 = 1000개 다른 이름을 가진 변수
  - 배열을 사용 할 경우 :
    - 1000개의 데이터 = 하나의 변수 이름과 + 번호, 반복문을 사용하여 데이터를 검색 할 수 있다.

배열

FUNERS

- 선언 :
  - 배열이름 = [ ] (비어있는 배열)
  - 배열이름 = [1,2,3,4,5,6]
- 배열에 데이터 추가 :
  - 배열이름.append(데이터)
- 배열의 데이터 읽기
  - 배열이름[번호]
- 배열의 길이:
  - 배열이름.__len__()
- 배열의 데이터를 크기순으로
  - 배열이름.sort()

배열

FUNERS

- 배열의 선언과 사용

```
[10, 123, 4, 5, 6, 7, 2, 4, 5, 13]
[10, 123, 4, 5, 6, 7, 2, 4, 5, 13, 10]
[2, 4, 4, 5, 5, 6, 7, 10, 10, 13, 123]
5
11
```

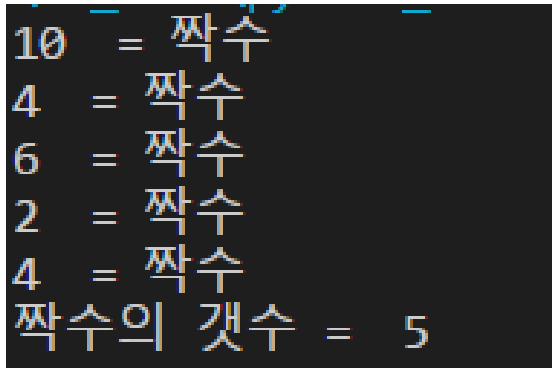
```
#선언
array = [10,123,4,5,6,7,2,4,5,13]
#데이터 출력
print(array)
#데이터 추가
array.append(10)
print(array)
#데이터 정렬
array.sort()
print(array)
#4번째 데이터 읽기
print(array[3])
#배열 크기
print(array.__len__())
```

반복문으로 배열 탐색

배열 사용하기

FUNERS

- 반복문으로 배열 탐색

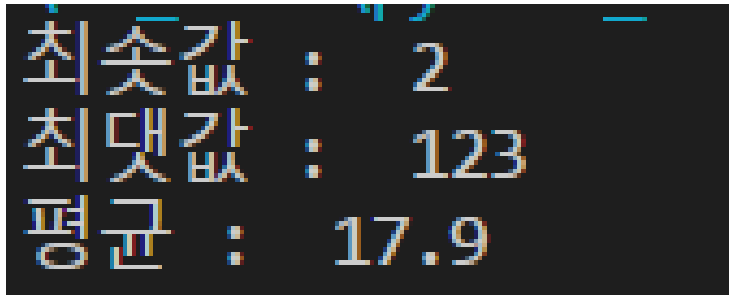



```
array = [10,123,4,5,6,7,2,4,5,13]

even_count = 0
for i in array :
    if i%2 == 0 :
        print(i, " = 짝수")
        even_count += 1

print("짝수의 갯수 = ", even_count)
```

최대/최소/평균값 구하기

배열 사용하기

- 배열의 최대, 최소, 평균 구하기




```
array = [10,123,4,5,6,7,2,4,5,13]
min_number = 1000
max_number = 0
avg_number = 0

for i in array :
    #최솟값 찾기
    if min_number > i :
        min_number = i
    #최댓값 찾기
    if max_number < i :
        max_number = i
    #평균 구하기1
    avg_number += i
#평균 구하기2
avg_number = avg_number/array.__len__()


print("최솟값 : ", min_number)
print("최댓값 : ", max_number)
print("평균 : ", avg_number)
```

2차 배열

2차 배열

FUNERS

- 2차 배열 – 아파트, 책장, 체스판
  - 배열이 들어있는 배열
  - 2개의 번호가 필요하다
    - 바깥쪽 배열에 저장된 배열들의 번호
    - 안쪽 배열에 저장된 데이터들의 번호
  - 선언 :
    - 배열이름 = [[1,2,3],[4,5,6],[7,8,9]]

디스플레이에 표시하기

디스플레이에 표시하기

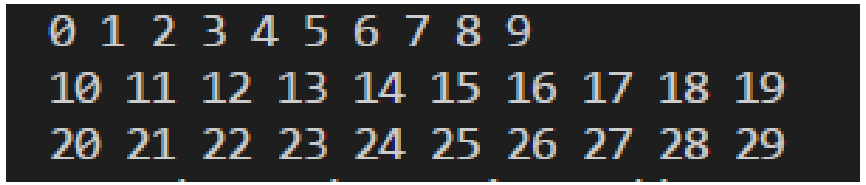
FUNERS

- 2차 배열 – 아파트, 책장, 체스판
  - 배열이 들어있는 배열
  - 2개의 번호가 필요하다
    - 바깥쪽 배열에 저장된 배열들의 번호
    - 안쪽 배열에 저장된 데이터들의 번호
  - 선언 :
    - 배열이름 = [[1,2,3],[4,5,6],[7,8,9]]

디스플레이에 표시하기

FUNERS

- 2차 배열을 아름답게 표시하기



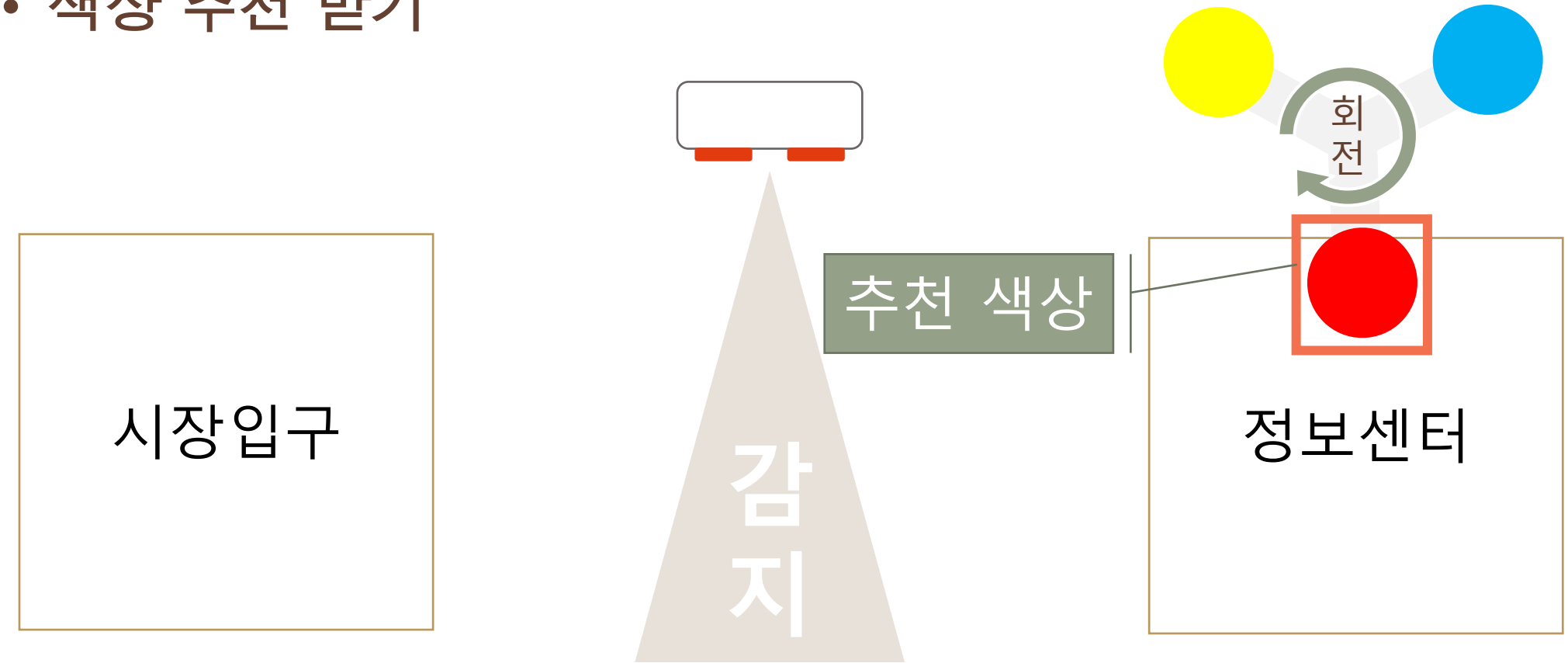

```
array = [[0,1,2,3,4,5,6,7,8,9],
         [10,11,12,13,14,15,16,17,18,19],
         [20,21,22,23,24,25,26,27,28,29]]
#lcd.clear() EV3에서 실행 할 때 활성화

for index1 in range(0,array.__len__()) :
    string = ""
    for index2 in range(0,array[index1].__len__()) :
        string += str(array[index1][index2])
        string += " "

    print(string)
```

2단계 미션

2단계 미션

FUNERS

- 색상 추천 받기



FUNERS

- **색상 추천 방법**
  - 시장 입구와 정보센터 사이의 빈 공간에 초음파 센서를 설치하여 시장 입구에서 정보센터로 이동하는 로봇을 감지 합니다.
  - 로봇의 이동이 감지되면 빨강, 노랑, 파랑 중 한가지 색상이 랜덤하게 선택 됩니다.
  - 색상은 3개의 날개를 가진 회전체의 각 날개 끝부분에 있습니다.
  - 회전체는 바닥에 붙어서 회전 하며 높이는 3mm ± 1 정도 입니다.

마무리

FUNERS

- 참고 사이트
  - EV3 Python 학습 사이트 : www.ev3python.com
  - API 참조 : http://python-ev3dev.readthedocs.io/en/latest/spec.html
  - Visual studio code 설치 관련: https://youtu.be/cqtRqsl6xMc
  - EV3와 PC의 연결 : https://youtu.be/TNXqizQTZhs

- 교육, 기술 지원 및 경기장 구매 문의

  퓨너스 (www.funers.com) T.070-8670-8911