

# 과제 개요

20171626 박태범

코드를 실행시키면 구동체가 장애물을 만날 때 까지 전진하고, 초음파를 통해 장애물을 감지하게 되면 일정거리 내에서 1차 과제는 오른쪽으로 스윙턴 후 다음 장애물을 만나게 되면 포인트턴을, 2차 과제는 왼쪽으로 포인트턴 후 다음 장애물을 만나게 되면 스윙턴을 해야 한다.

# 요구사항 분석

20171626 박태범

1. 주행 시 전방 장애물을 인식하고 회피해야 한다.
2. 1차 과제는 장애물 인식 시 우회전하고 2차 과제는 장애물 인식 시 좌회전해야 한다.
3. 회전 각도가 90도여야 한다.
4. 장애물 10cm 근방에서 정지한다.
5. 두번 회전이 끝난 후 3초간 전진하고 정지하여야 한다.

# 요구 기능

20171626 박태범

1. 하드웨어 초기화
2. 직진 기능
  - 전진, 후진
3. 장애물 인식 기능
4. 속도 조절기능
  - 왼쪽, 오른쪽 바퀴
5. 회전 기능
  - 스윙턴, 포인트턴
6. 동작시간 조절

# 프로그램 설계

20171626 박태범

함수 제작에 앞서 구동체를 작동시키기 위해 먼저 하드웨어의 초기값을 지정해 주었다. 그 후 왼쪽 모터와 오른쪽 모터에 True값을 입력 받으면 양쪽 모터가 시계방향(직진)으로 회전하도록 만들었고 False값을 입력 받으면 반시계방향(후진)으로 회전을 하도록 만들었다. 전진은 스피드만 입력받아 계속 작동하게 만드는 go\_forward\_any와 스피드와 시간을 입력받아 정해진 시간동안 전진시키는 go\_forward함수 두가지를 만들었다. 또한 회전을 위해 원하는 회전방향의 반대편 바퀴를 시계방향으로 돌게 해 구동체를 회전시키는 포인트턴, 한쪽 바퀴는 시계방향, 한쪽 바퀴는 반시계방향으로 돌게 해 구동체를 회전시키는 스윙턴을 구현하였다. 앞에 있는 장애물과의 거리를 초음파센서로 음파를 보내고 받는 시간을 2로 나누고 음속을 곱해 계산하였다.

# 프로그램 구현 및 평가 결과

20171626 박태범

`dis = 30` | 구동체의 속도를 고려해 멈출 거리를 30으로 지정하였다.

```
try:
    while True:
        distance = getDistance()
        print('distance= ', distance)
        go_forward_any(40)
```

while 문을 이용해 초음파 센서를 이용해 구한 거리가 임의로 정한 거리보다 작으면 계속 직진하도록 구현하였다.

```
if (distance < dis):
    stop()
    time.sleep(1)
    leftPointTurn(22,0.3)
    time.sleep(1)
```

구한 거리가 임의로 정한 거리보다 작으면 우선 멈추고 leftPointTurn 을 하도록 만들었다.

```
while True:
    distance2 = getDistance()
    print('distance2= ', distance2)
    go_forward_any(40)
    if (distance2 < dis):
        stop()
        time.sleep(1)
        leftSwingTurn(25,0.4)
        time.sleep(1)
        go_forward(40,3)
        quit()
```

반복문을 하나 추가해 다음 장애물과의 거리를 계속 측정하고 임의로 지정한 거리보다 멀다면 계속 직진하도록 만들었다.

두번째로 구한 거리가 임의로 정한 거리보다 작으면 잠시 멈추고 leftPointTurn 을 하게 하도록 하였다. 그 다음 3초간 앞으로 전진 후 함수가 종료된다.

```
except KeyboardInterrupt:
    pwm_low()
```

Keyboardinterrupt 에러가 나면 구동체를 정지시키는 try except 문이다.

# 결론 및 느낀점

20171626 박태범

## 1. 회전 각도와 속도는 환경의 영향을 굉장히 많이 받는다.

예) 배터리의 잔여량, 바닥의 종류(같은 곳이라도 위치에 따라 달라지기도 함)

## 2. 전선(?)을 잘 꽂아야 한다.

당연한 이야기이지만 막상 한두개씩 잘못 꽂는 경우가 있다. 초음파 센서가 작동하지 않는다면 우선 선부터 잘 확인하는 것이 좋다.

## 3. 모터의 회전 방향은 다를 수 있다.

예시 코드는 첫번째 핀과 연결된 부분을 HIGH, 두번째 핀과 연결된 부분을 LOW로 설정하면 바퀴가 시계방향(전진)을 한다고 되어있지만 모두 그런 것은 아니다. 혹시 바퀴가 반시계방향(후진)을 하고 있다면 코드의 HIGH와 LOW의 위치를 바꿔보자

이번 과제는 사용자가 구동체의 움직임을 구체적으로 정한 것이 아니라 주변의 입력을 받아 스스로 움직임을 변화시켰다는데 의의가 있다고 생각합니다. 정확히 90도를 돌기 위해 미세한 수치 조정을 하며 수십번 구동체를 작동시키는 것 말고 다른 방법으로 평가받을 수 있었다면 더 좋았을 것 같다는 생각이 들었습니다.