

DU-도전학기 결과보고서

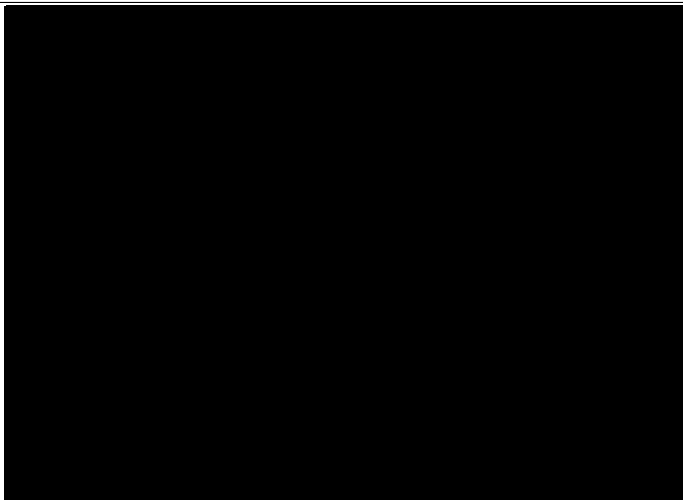
과제명	ROS를 활용한 실내 자율주행 로봇 제작		
참여자	성명	소속	학번
	■	전기공학전공	■
	김 ■	전기공학전공	■
	이 ■	전기공학전공	■
지도교수 의견	이 ■	전기공학전공	■
	새로운 자율주행 S/W 플랫폼 공부를 통한 알고리즘 구현 및 맵구성 H/W를 전체 시스템 완성으로 높은 목표 도전 및 실현 우수 (소속) 전기공학전공 (성명) ■		

1. 도전 과제 내용

ROS 학습 및 ROS를 활용해 실내 자율 주행 로봇 제작

2. 도전 과제 수행 결과 및 성과

- ROS 개념 및 원리 세미나 (한 ■)

			
ROS 세미나			

- ROS를 통해 아두이노에서 노드 값 송신 및 아두이노 제어()

```

void cmd_vel_callback( const geometry_msgs::Twist& msg){
    M1_motor = int(msg.linear.x);
    M2_motor = int(msg.angular.z);
}

ros::Subscriber<geometry_msgs::Twist> cmd_sub("main", cmd_vel_callback);
ros::Publisher M1_pub("M1", &M1_msg);
ros::Publisher M2_pub("M2", &M2_msg);
ros::Publisher value_pub("value", &value_msg);
        
```

아두이노 subscriber

- Lidar 센서 값 분석을 통해 각도, 거리 산출, 장애물 탐지, 노드 값 publish()

```

def scanToPoint(radian, distance):
    global x, y
    x = distance * numpy.cos(radian)
    y = distance * numpy.sin(radian)
    return x, y

def callback(data):
    global x, y, degree, ranges, M1_motor, M2_motor
    count = int(data.scan_time / data.time_increment)
    for i in range(0, count, 1):
        degree = numpy.rad2deg(data.angle_min + data.angle_increment * i)
        radian = data.angle_min + data.angle_increment * i
        scanToPoint(radian, data.ranges[i])
        ranges = data.ranges[i]
        
```

각도:34.000014997827456, 거리:2.3429999351501465
 각도:35.00001499008109, 거리:1.9869999885559082
 각도:36.000014982334726, 거리:2.5320000648498538
 각도:37.00001497458836, 거리:2.6600000858306885
 각도:38.00001496684199, 거리:2.734999985095825
 각도:39.00001495909562, 거리:2.430999994277954
 각도:40.00001495134926, 거리:2.4079999923706055
 각도:41.00001494360289, 거리:2.388000011444092
 각도:42.00001493585653, 거리:2.4100000858306885
 각도:43.000014928110154, 거리:2.377000093460083
 각도:44.00001492036379, 거리:2.3499999946325684
 각도:45.000014912617424, 거리:2.3299999237060547

Click/Mouse Wheel: Zoom. Shift: More options. 31 fps

Lidar publish

- meanshift 알고리즘을 사용해 객체 추적()

```

if histogram is not None:
    # HSV space로 변환
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
    # 역투영
    dst = cv2.calcBackProject([hsv], [0], histogram, [0,180], 1)
    # 평균 이동 추적
    ret, (x,y,w,h) = cv2.meanShift(dst, (x,y,w,h), terminal)
    cv2.rectangle(draw,(x,y), (x+w, y+h), (0,255,0), 2)
    XX = (x+x+w)/2
    YY = (y+y+h)/2
    cv2.putText(draw, "x = " + str(XX), (10,30), cv2.FONT_HERSHEY_SIMPLEX, 1, (0,0,255), 1, cv2.LINE_AA)
    cv2.putText(draw, "y = " + str(YY), (10,60), cv2.FONT_HERSHEY_SIMPLEX, 1, (0,0,255), 1, cv2.LINE_AA)
    if XX > W/2+100:
        print("right")
    elif XX < W/2-100:
        print("left")
    # 추적 영상 및 역투영 영상 동시 출력
    # result = np.hstack((draw, cv2.cvtColor(dst, cv2.COLOR_GRAY2BGR)))
    result = draw
        
```

meanshift 알고리즘

- 메인 알고리즘 개발 ()

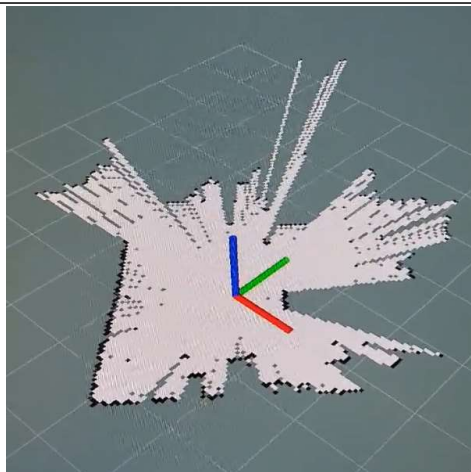
```
pubmsg = Twist()
if Stering == "right":
    M1_motor = 0
    M2_motor += 20
elif Stering == "left":
    M1_motor += 20
    M2_motor = 0
# elif Stering == "str":
else:
    #if (degree <= 200) or (degree >= 150):
    print(degree, ranges, M1_motor, M2_motor, type(ranges))
    if (ranges >= 50) and (ranges <= 10000):
        M1_motor += 10
        M2_motor += 10
    else:
        M1_motor = 0
        M2_motor = 0
M1_motor = 0
M2_motor = 0

pubmsg.linear.x = M1_motor
pubmsg.angular.z = M2_motor
publisher.publish(pubmsg)
```

```
sub_scan = rospy.Subscriber('/scan', LaserScan, callback = process_lidar)
sub_cam = rospy.Subscriber('/cam', String, callback = process_cam)
```

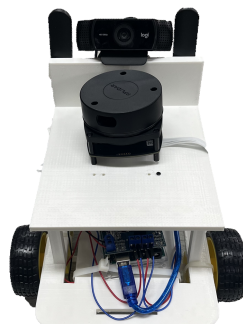
메인 알고리즘

- SLAM을 통해 실내 지도 생성 (이 , 이)



SLAM mapping

- 인벤터를 사용해 3D프린팅 출력 (김)



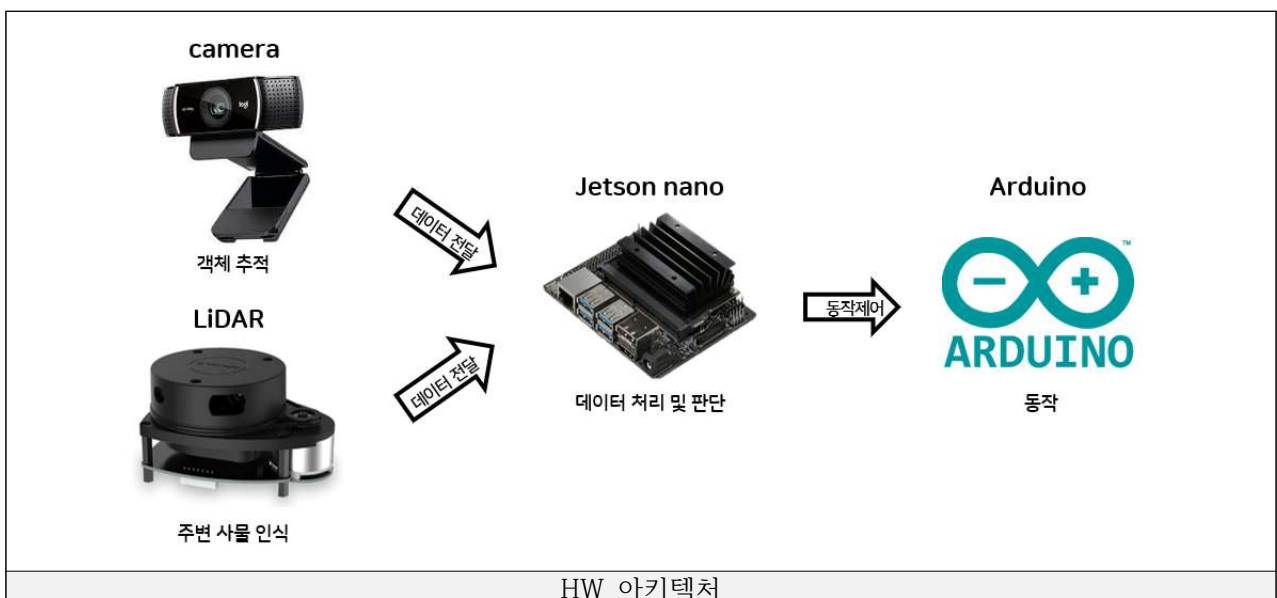
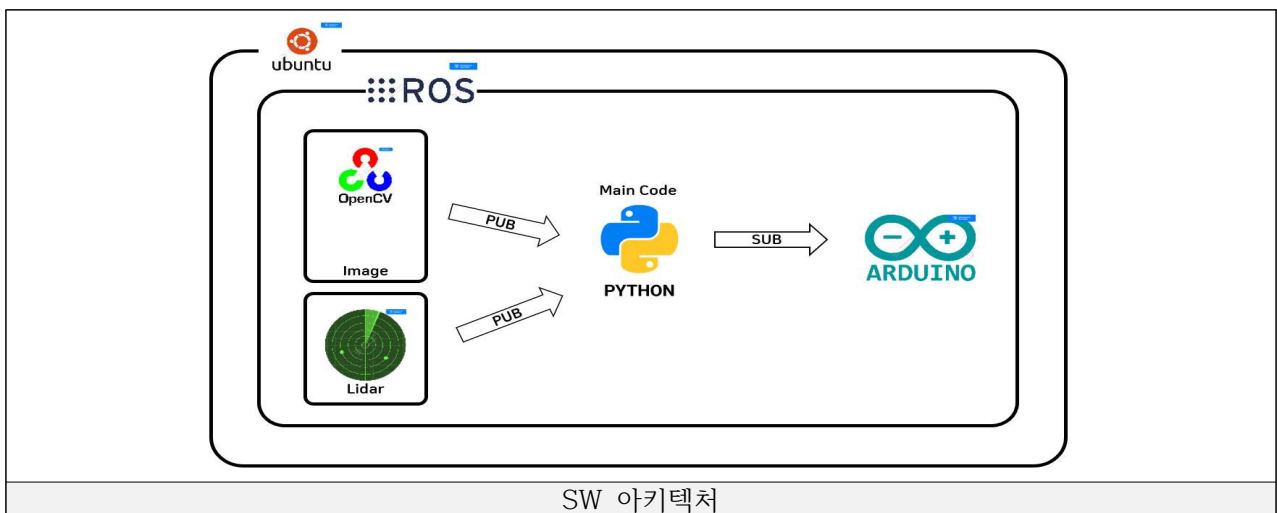
3d 프린팅

3. 자기 평가

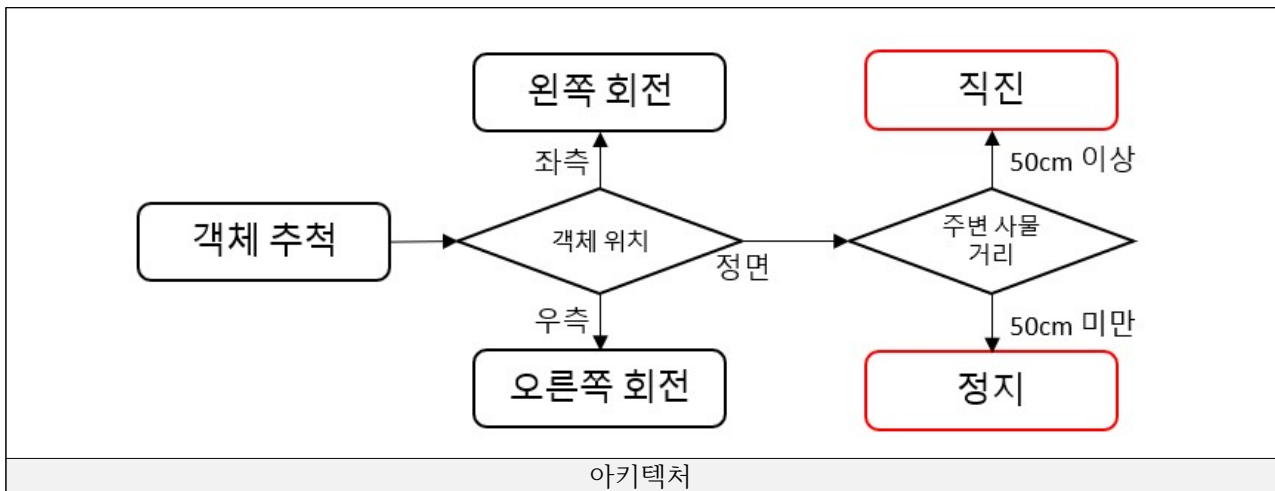
이름	평가
한 [REDACTED]	처음으로 진행한 팀 프로젝트라 팀장으로서 역할분배나 일정 조율, 개발 등 전반적인 부분에서 많이 부족했던 것 같다. 다음에 또 이런 기회가 생겨 팀 프로젝트를 진행한다면 지금의 경험을 바탕으로 팀원들과 더욱 효과적으로 프로젝트를 진행할 수 있도록 노력할 것이다.
김 [REDACTED]	이런 개발을 처음 진행해봐서 많이 부족하단걸 느꼈고 이번 프로젝트 경험을 통해 남은 대학생활동안 열심히 공부해서 다른 작품이나 졸업 작품을 제작할 때 많은 도움이 될 것 같다.
이 [REDACTED]	평소 임베디드, 자율주행과 같은 분야에 관심이 많았는데 도전학기라는 좋은 기회로 나만의 작품을 만들 수 있었고 처음 생각한 것 보다 완성도 높은 작품이 제작되어 좋았다.
이 [REDACTED]	객체추적이란 분야를 처음 접해봤는데 생각보다 재미있었고 지금은 좀 오래된 기술인 meanshift 기술을 사용했지만 다음에 기회가 된다면 더 좋은 기술을 사용하고 튜닝하여 더욱 완성도 있는 작품을 만들고싶다.

4. 최종 결과물

- 아키텍처



- 플로우차트



- 결과물 사진

