



# 서봇

자율주행서빙로봇

척척박사

20141118 - 전수현

20151095 - 김은영



# CONTENTS

---

개발 필요성 및 목적

진 행 상 황

T F

모 터 제 어

M A P 생 성

N a v i g a t i o n

영 상 처 리

개 발 일 정

## 개발 필요성 및 목적



1. 최저임금 인상

사업자들의 비용절감 효과

2. 비대면 서비스 증가  
(Un-Contact)

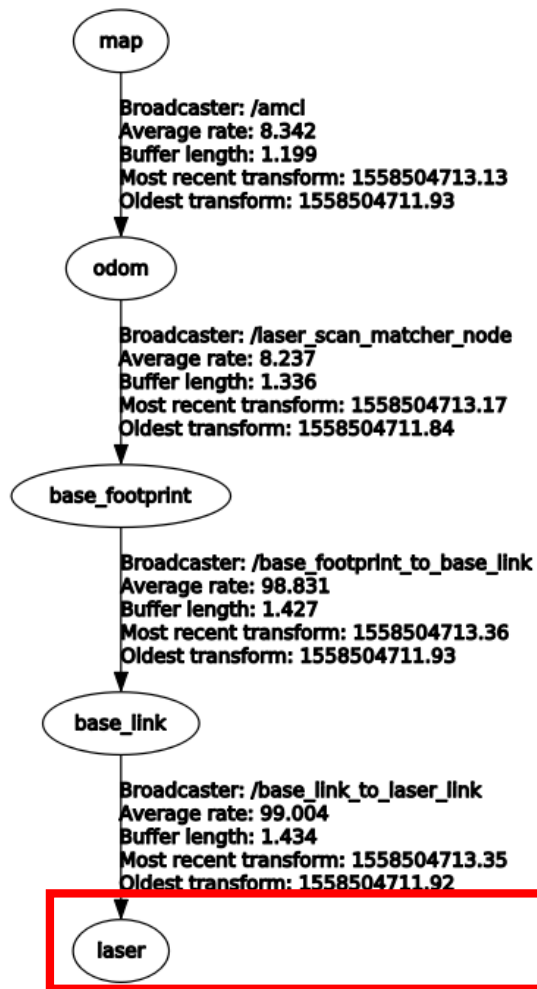
비대면 서비스 선호 및  
심리적 편안함

3. 단순 업무 대체

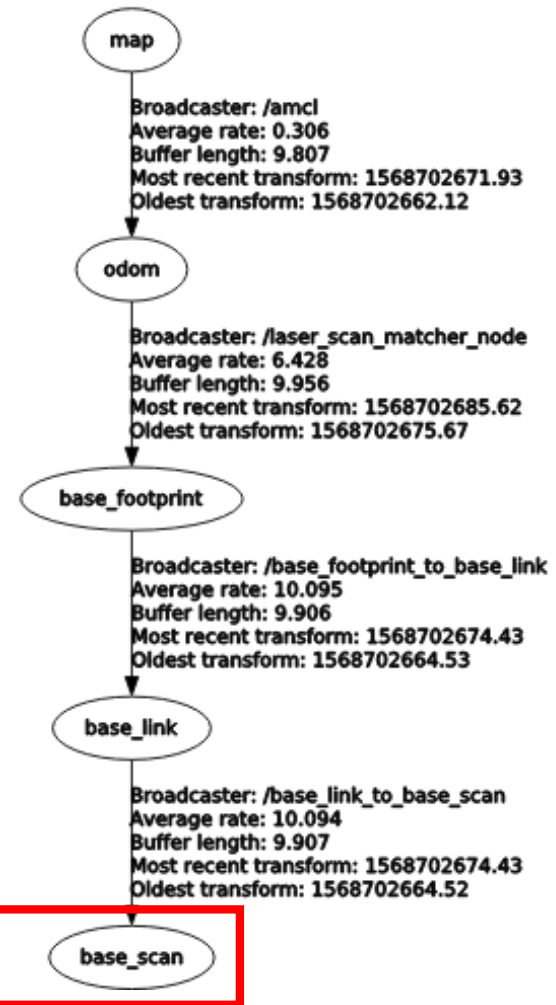
업무 처리 시간 감소 및  
소비자들의 대기시간 감소



## 진행상황 - TF



기존 TF



새로운 TF

# | 진행상황 - TF



Serbot\_tf.launch

```
<?xml version="1.0" encoding="utf-8"?>

<launch>

  <!-- ***** TFs ***** -->
  <arg name="pi" value="3.14159265358979" />
  <arg name="pi/2" value="1.5707963267948966" />
  <arg name="pi/4" value="0.785398163397448" />
  <arg name="optical_rotate" value="0 0 0 -$(arg pi/2) 0 -$(arg pi/2)" />
  <arg name="imu_rotate" value="0 0 0 -$(arg pi/2) 0 0" />

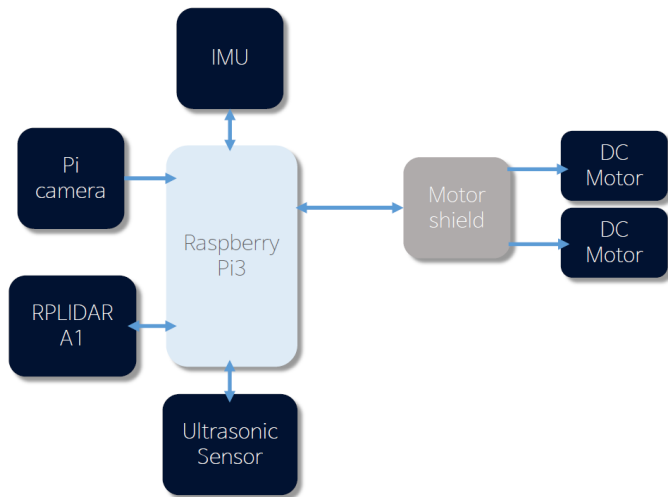
  <!-- base footprint to base link -->
  <node pkg="tf" type="static_transform_publisher" name="base_footprint_to_base_link"
    args="0 0 0 0 0 base_footprint base_link 10"/>

  <node pkg="tf" type="static_transform_publisher" name="base_link_to_base_scan"
    args="0 0 0 0 0 base_link base_scan 10"/>

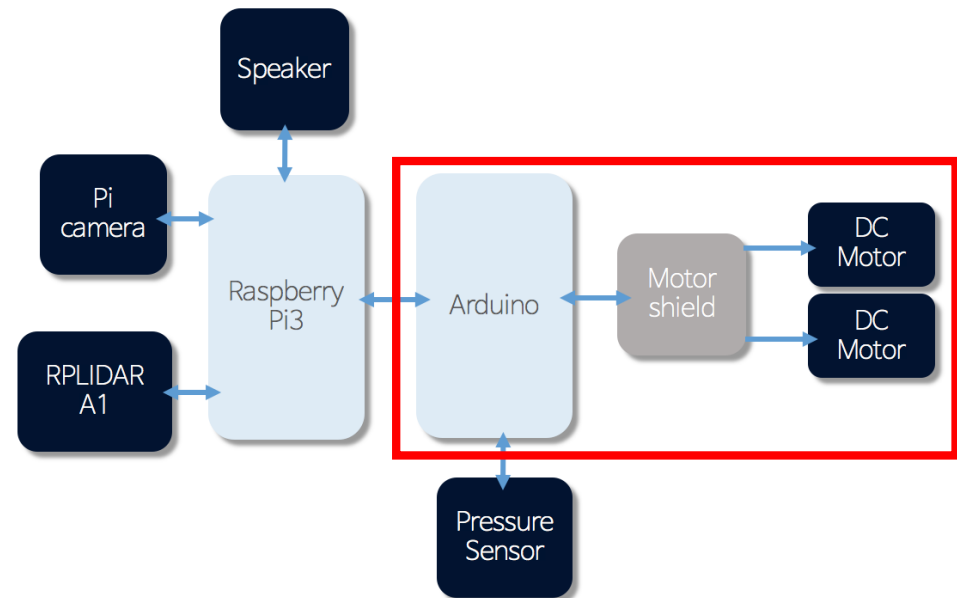
</launch>
```



## 진행상황 - 모터 제어



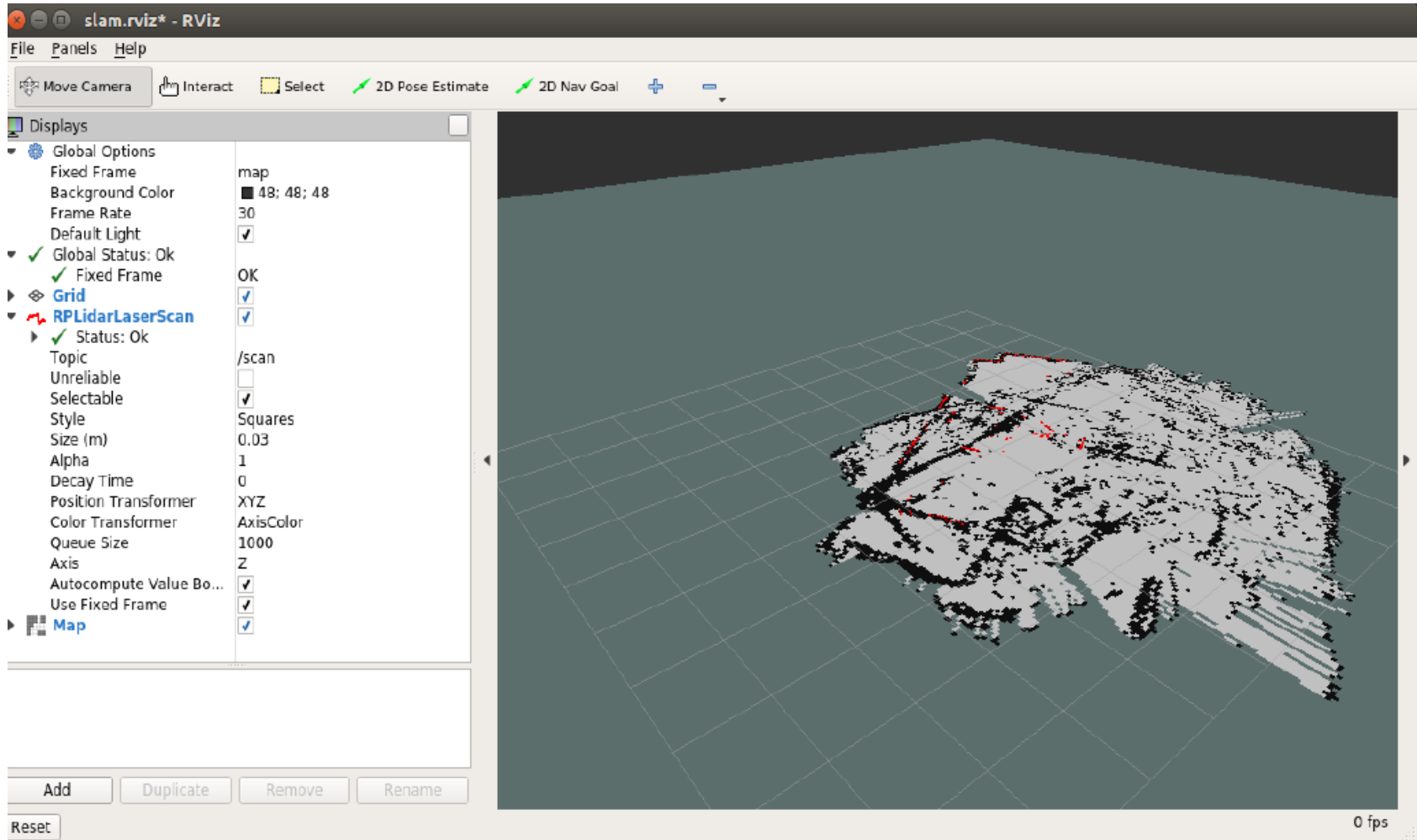
기존 서봇 하드웨어 구성도



새로운 서봇 하드웨어 구성도



## 진행상황 - MAP생성

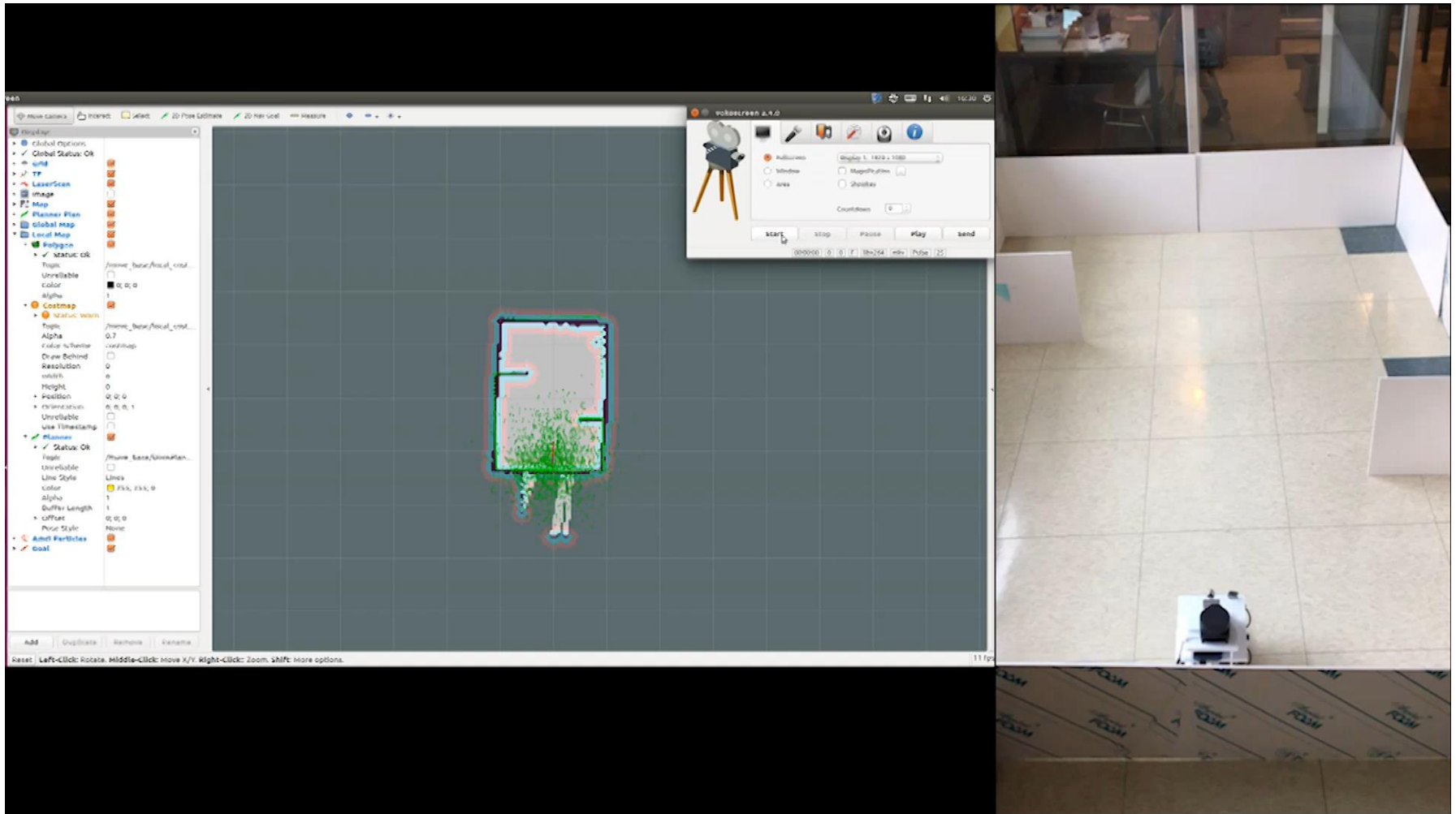


이전 문제 상황





# 진행상황 - Navigation



마우스로 위치 지정

# 진행상황 - Navigation



Serbot\_return.py

```
def movebase_client():

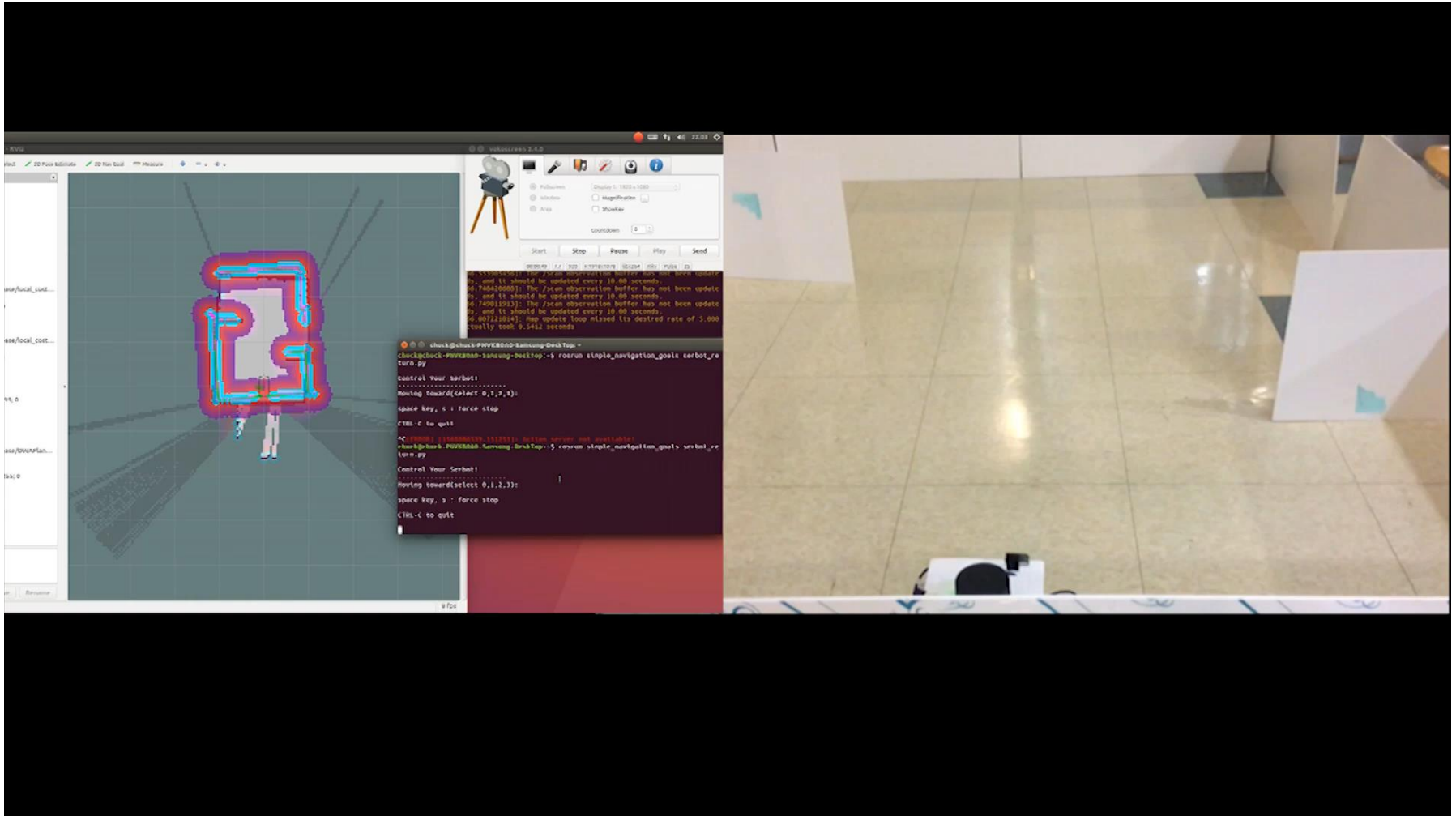
    client = actionlib.SimpleActionClient('move_base', MoveBaseAction)
    client.wait_for_server()

    goal = MoveBaseGoal()
    goal.target_pose.header.frame_id = "map"

    try:
        print msg
        while(1):
            key = getKey()
            print key
            if key == '0' :
                goal.target_pose.header.stamp = rospy.Time.now()
                goal.target_pose.pose.position.x = 0.5
                goal.target_pose.pose.position.y = 0.0
                goal.target_pose.pose.position.z = 0.0
                goal.target_pose.pose.orientation.w = 1.0
                client.send_goal(goal)
            elif key == '1' :
                goal.target_pose.header.stamp = rospy.Time.now()
                goal.target_pose.pose.position.x = 1.0
                goal.target_pose.pose.position.y = 0.0
                goal.target_pose.pose.position.z = 0.0
            elif key == ' ' or key == 's' :
                pub_stop()
            else:
                if (key == '\x03'):
                    break
```



# 진행상황 - Navigation



코드로 위치 지정



## 진행상황 - 영상처리

Serbot\_fullbody.launch

```
<launch>
  <node pkg="serbot_opencv" type="serbot_fullbody_detect.py" name="serbot_fullbody_detect" output="screen">
  </node>

  <node pkg="serbot_opencv" type="serbot_fullbody_listener.py" name="serbot_fullbody_listener" output="screen">
    <param name="tts_file" value="/home/chuck/Downloads/1.mp3" />
  </node>
</launch>
```

Serbot\_fullbody\_detect.py

```
def main(args):
    rospy.init_node('serbot_fullbody_detect', anonymous=True)

    image_sub = rospy.Subscriber("/camera/image", Image, callback)

    try:
        rospy.spin()
    except KeyboardInterrupt:
        print("Shutting down")
        cv2.destroyAllWindows()

if __name__ == '__main__':
    main(sys.argv)
```



## 진행상황 - 영상처리

Serbot\_fullbody\_detect.py

```
# fullbody is not empty
if fullbody != None and len(fullbody) > 0:
    rospy.loginfo('fullbody detected: %s, started %s took %s' % (str(fullbody), t1, rospy.get_time() - t1))
    pub.publish(str(fullbody))
    stop_pub.publish(str("STOP"))

for (x,y,w,h) in fullbody:
    cv2.rectangle(image,(x,y),(x+w,y+h),(255,0,0),3)
# show the frame
#cv2.imshow("Frame", image)
try:
    image_pub.publish(bridge.cv2_to_imgmsg(image, "bgr8"))
except CvBridgeError as e:
    print(e)
```

Serbot\_fullbody\_listener.py

```
pub = rospy.Publisher('/play_sound_file', String, queue_size=10)
tts_param = rospy.get_param('tts_file', '/home/chuck/Downloads/1.mp3')
```



사람 인식



## 9월 & 10월

1. 하드웨어 교체
2. 무게 센서
3. 디버깅



감사합니다