

창의 자율과제 보고서

과제명	이동로봇에 대한 모니터링 시스템 개발
기간	2020.10.1. ~ 2020.11.30
내용	이동로봇을 원격에서 로봇을 관리(이동경로/영상, 상태, 제어) 할 수 있는 모니터링 시스템을 개발하여, 1대 또는 다량의 로봇을 안전하게 운영 및 서비스
참여인력	멘토 : 임영태 박사
	멘티 : 석사과정 임동민, 차상협
연구목표	<ul style="list-style-type: none"> - 로봇과 서버간 전송 데이터를 정의 한다. - 로봇과 서버간 통신 프로토콜을 설계 및 개발 한다. - 서버에서 데이터 표출방식을 설계 및 개발한다.
연구의 중요성	<p>로봇의 이동패턴과 상태를 분석하여 응용 설계에 활용 가치가 있음. 로봇에서 발생하는 데이터를 표준화된 데이터를 정의하고 프로토콜화하여 다량, 이기종 로봇의 정보를 통합서버(운영서버)에서 데이터를 수집하고, 로봇을 모니터링, 빅데이터화, 데이터가공, 오픈API 제공, 로봇의 플랫폼화 사업으로 확장하는데 기초 기술로 활용 가치가 있음.</p> <p>로봇의 컴퓨팅파워에서 센서들의 인지판단과정을 로컬에서 센서의 raw data를 활용하여 시각화하는데 별문제가 없지만, 서버에서 시각화 하려면 대량을 센서데이터를 수신해야하는데 네트워크 제약의 이슈로 시각화 하는데 제약사항이 발생된다. 그래서 로컬에서 다양하게 분석된 시각화 영상정보를 오픈 VNC로 전송받아서 표출하는 방식을 고려 함.</p> <p>현재 무선통신망(WiFi/LTE/5G) 네트워크 사항을 고려한 통신망 선정, 데이터 밴드위스 고려한 데이터를 정의하는데 활용</p>
연구내용	<p>1. 시스템 구성</p> <p style="text-align: center;">로봇 서버 데이터 연계 관계도</p> <p style="text-align: center;"> 이동로봇 모니터링 서버 </p> <p>로봇 : 터틀봇3 와올파이 + 라즈베리파이4 + LCD 서버 : 노트북(OS-Ubuntu) + 와이드 모니터 통신 : Kafka framework(TCP/IP), RTSP, VNC</p>

2. (10.01~10.31) 로봇환경 Setup

- 터틀봇3 와플파이 성능 Upgrade를 위한 기본 내장된 라즈베리파이3+를 라즈베리파이4+로 교체

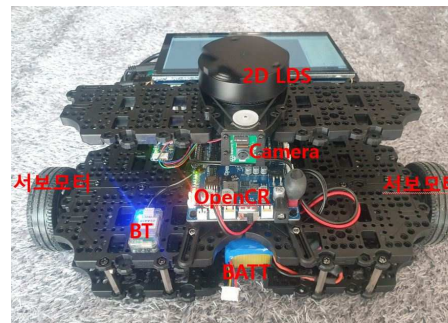
특징 / 사양	라즈베리 파이 4B	시스템 사양 Upgrade	라즈베리 파이 3 B+	터틀봇3 기본 사양
출시일	2019년 6월 24일		2018년 3월 14일	
SoC	Broadcom BCM2711 쿼드 코어 Cortex-A72 @ 1.5GHz	CPU 성능 향상	Broadcom BCM2837B0 1.4GHz 쿼드 코어 Cortex-A53	
GPU	OpenGL ES 1.1, 2.0, 3.0이 포함 된 VideoCore VI	GPU 성능 향상	OpenGL ES 1.1, 2.0이 포함 된 VideoCore IV	
비디오 디코딩	H.265 4Kp60, H.264 1080p60		H.264 및 MPEG-4 1080p30	
비디오 인코딩	H.264 1080p30			
기억	1GB, 2GB 또는 4GB LPDDR	RAM 속도 향상, 8G 용량 UP	1GB LPDDR2	
저장	microSD 카드	USB3.0 Memory 256G, boot 지원		SD Boot
비디오 및 오디오 출력	최대 4Kp60의 2x 마이크로 HDMI 포트 3.5mm AV 포트 (컴포지트 + 오디오) MIPI DSI 커넥터		1x 마이크로 HDMI 1.4 포트 최대 1080p60 3.5mm AV 포트 (컴포지트 + 오디오) MIPI DSI 커넥터	
카메라	MIPI CSI 커넥터			
이더넷	기본기가비트 이더넷		USB를 통한기가비트 이더넷 (최대 300Mbps)	
와이파이	듀얼 밴드 802.11 b/g/n/ac			
블루투스	블루투스 5.0 + BLE		Bluetooth 4.2 + BLE	
USB	2x USB 3.0 + 2x USB 2.0		4x USB 2.0	
확장	40 핀 GPIO 헤더			
전원 공급 장치	USB 유형 -C를 통해 5V 최대 3A까지 3A GPIO 헤더를 통해 5V PoE HAT를 통한 PoE (Power over Ethernet)		마이크로 USB를 통해 최대 2.5A의 5V GPIO 헤더를 통한 5V 최대 3A PoE HAT를 통한 PoE (Power over Ethernet)	

Display

7", 1024*768, IPS

- 개발환경

- 터틀봇3 와플파이 조립



앞면 촬영 사진



위에서 촬영 사진

3. (10.15~11.15) 로봇과 서버 환경 및 기능 검증

- 로보 운영 S/W
 - 터틀봇3 와플파이
 - OS : Ubuntu 18.04
 - ROS : melodic
 - OpenCR : 아두이노 프로그램(모터제어, IMU 센싱, 파워공급 등)

- S/W 패키지 설치 및 기능 검증

[라즈베리파이4 부팅 USB에 OS 설치]

```
ubuntu-mate-18.04.5 install
```

```
-----  
zcat vmlinuz > vmlinux
```

```
sudo chmod +x auto_decompress_kernel
```

```
// /ect/apt/apt.conf.d/
```

```
sudo chmod +x 999_decompress_rpi_kernel
```

```
-----  
sudo apt update
```

```
sudo apt upgrade
```

```
sudo apt install ubuntu-mate-desktop
```

[ROS setup]

```
-----  
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu  
$(lsb_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'
```

```
sudo apt-key adv --keyserver 'hkp://keyserver.ubuntu.com:80'  
--recv-key C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654
```

```
sudo apt update
```

```
sudo apt install ros-melodic-desktop-full
```

```
sudo apt-get install ros-kinetic-desktop-full
```

```
sudo apt install python-pip
```

```
sudo pip install -U rosdep
```

```
sudo rosdep init
```

```
rosdep update
```

```
mkdir -p ~/catkin_ws/src
```

```
cd ~/catkin_ws
```

```
echo "source /opt/ros/melodic/setup.bash" >> ~/.bashrc
```

```
echo "source /opt/ros/kinetic/setup.bash" >> ~/.bashrc
```

```
source ~/.bashrc
```

```
catkin_make
```

[TurtleBot3]

```
-----  
# edit .bashrc
```

```
source ~/catkin_ws/devel/setup.bash
```

```
export ROS_MASTER_URI=http://172.30.1.17:11311
```

```
export ROS_HOSTNAME=172.30.1.50
```

```
export TURTLEBOT3_MODEL=waffle_pi
```

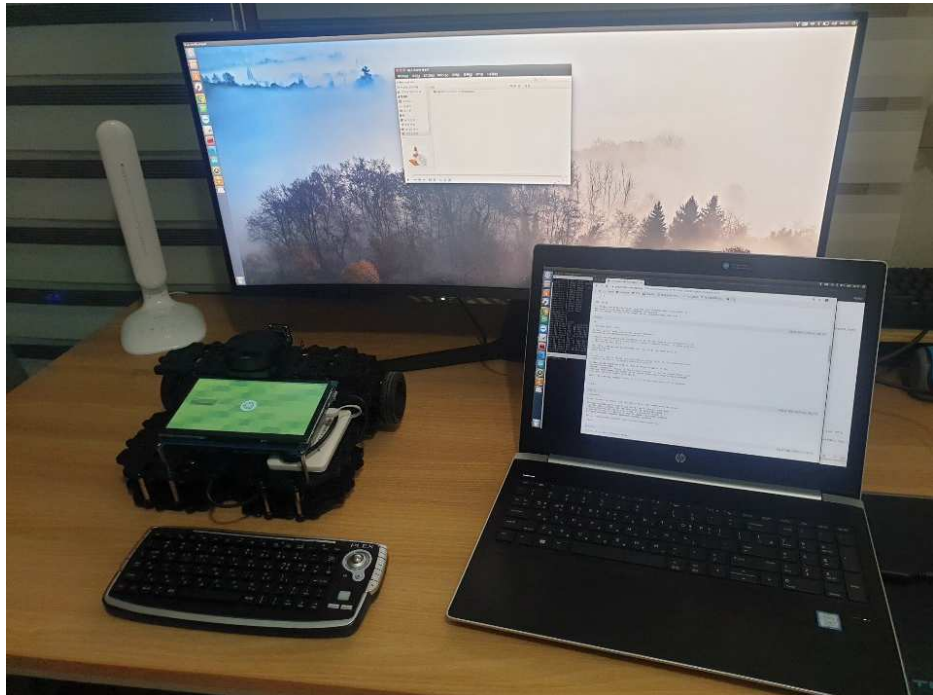
```

source ~/.bashrc
[TurtleBot]
cd ~/catkin_ws/src
git clone https://github.com/ROBOTIS-GIT/hls_lfcd_lds_driver.git
git clone https://github.com/ROBOTIS-GIT/turtlebot3_msgs.git
git clone https://github.com/ROBOTIS-GIT/turtlebot3.git
$ cd ~/catkin_ws/src/turtlebot3
$ rm -r turtlebot3_description/ turtlebot3_teleop/
  turtlebot3_navigation/ turtlebot3_slam/ turtlebot3_example/
sudo apt-get install ros-kinetic-rosserial-python ros-kinetic-tf
#source /opt/ros/melodic/setup.bash
#source /opt/ros/kinetic/setup.bash
cd ~/catkin_ws && catkin_make -j1
roslaunch turtlebot3_bringup create_udev_rules
-----
roscore
roslaunch turtlebot3_bringup turtlebot3_robot.launch
roslaunch turtlebot3_bringup turtlebot3_remote.launch
roslaunch rviz rviz -d `rospack
  find turtlebot3_description`/rviz/model.rviz
roslaunch turtlebot3_teleop turtlebot3_teleop_key.launch
roslaunch turtlebot3_slam
  turtlebot3_slam.launch slam_methods:=gmapping
$ roslaunch turtlebot3_bringup turtlebot3_lidar.launch
$ roslaunch turtlebot3_bringup turtlebot3_rpicamera.launch
$ roslaunch turtlebot3_bringup turtlebot3_realsense.launch
$ roslaunch turtlebot3_bringup turtlebot3_core.launch

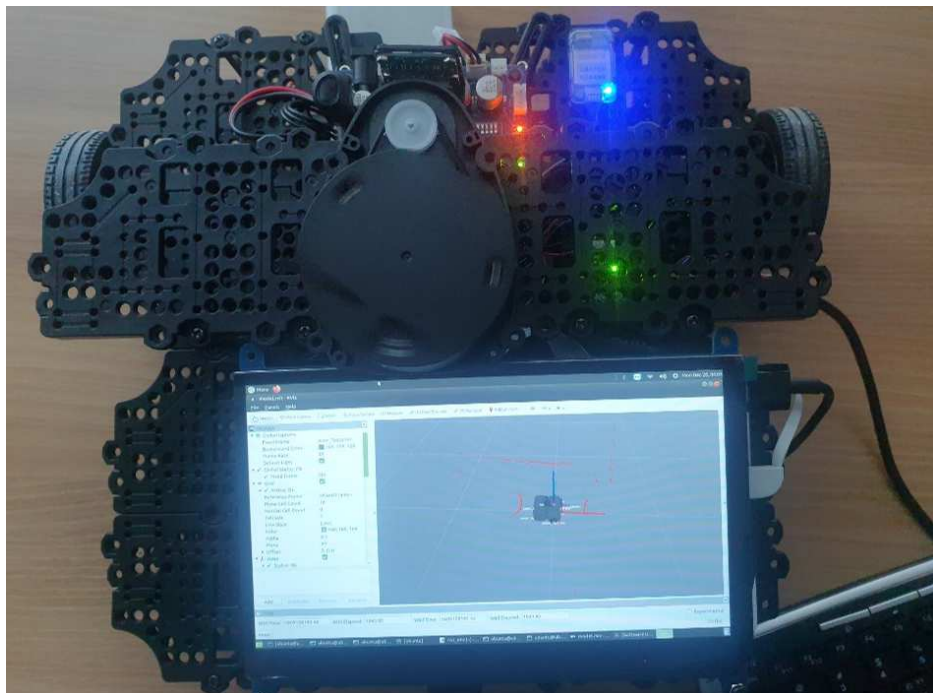
[ VNC ]
sudo apt-get install lightdm
sudo apt-get install x11vnc
x11vnc

```

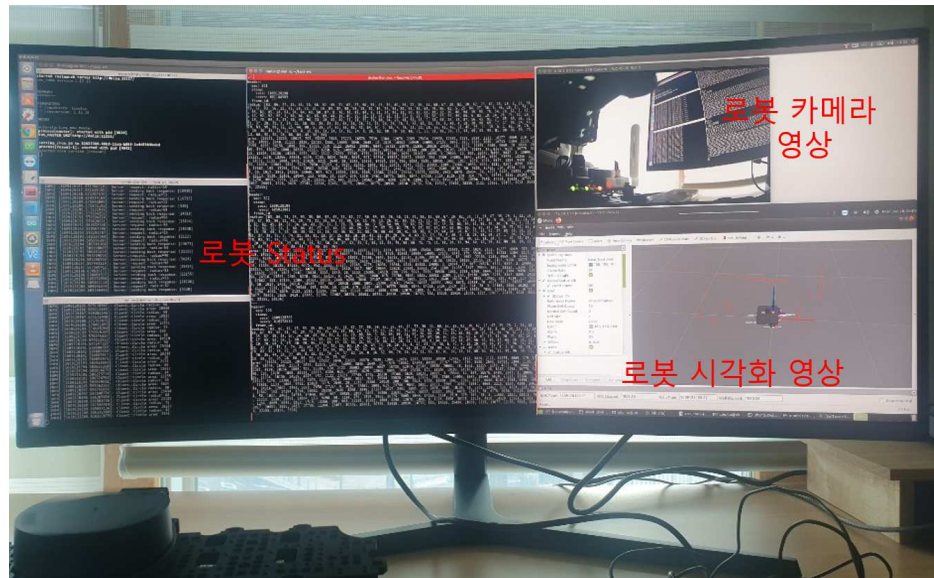
4. (11.16~11.30) 기능 통합 및 검증 - 결과
- 로봇과 서버 개발 시스템 환경



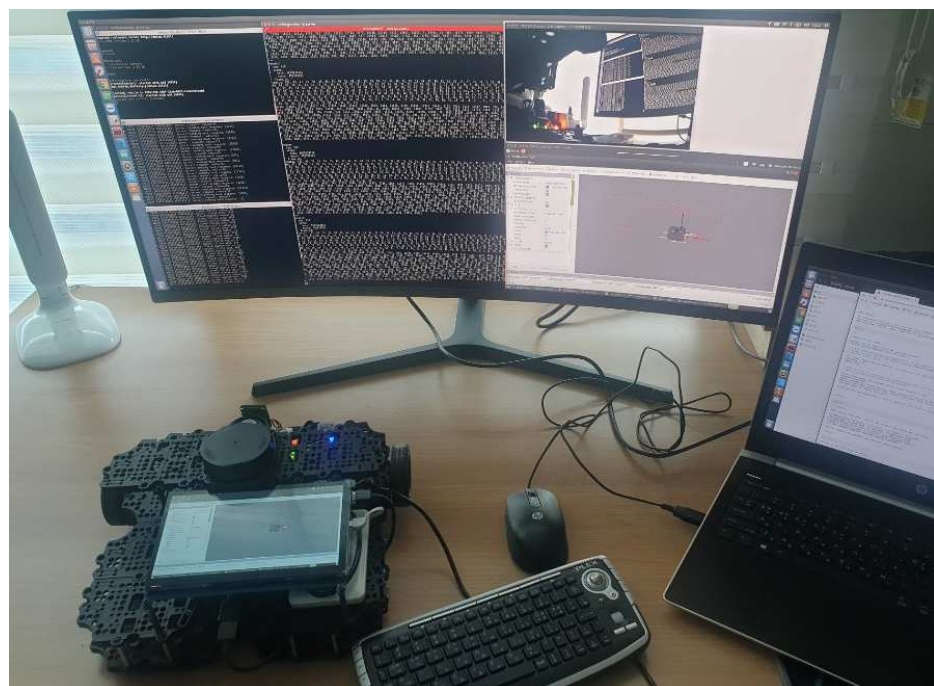
- 로봇(티틀봇3) 실행화면
- RVIZ로 로봇의 LDS/IMU 센서, Motor 움직임 및 이동량에 대한 시각화영상 Display(7" LCD)
- 시각화영상을 VNC로 서버에 전송



- 로봇의 UVC Camera를 RTSP로 서버에 전송
- 로봇의 이동상태를 ros message로 서버에 전송
- 서버에서 로봇의 정보를 모니터링
 - 로봇의 시각화영상, 카메라영상, 로봇상태값을 전송받아 서버의 모니터링 화면에 Display



- 이동로봇에 대한 모니터링 시스템 통합 결과



추진체계	구분	2020. 10	2020.10	2020.11	2020.11
	로봇환경 Setup				
	로봇과 서버 개발환경				
	기능별 단위 검증				
	기능 통합 및 검증				
	주요 Milestone 수행결과				
실적	<ul style="list-style-type: none"> ■ 시제품 ■ 학술발표 ■ 논문 				