

## 0. 작성 시 주의사항

※아래의 작성 양식(제출분량, 폰트, 크기, 줄 간격 등)을 미준수 시 서류 평가의 감점요인됨

- ※ 제출 분량 : A4 용지 상세내용 포함 30 page 이내
- ※ 작성 양식 (폰트 : 맑은 고딕 / 폰트 크기 : 10pt / 자간 : 0% / 장평 : 100% / 줄 간격 : 130%)
- ※ 제출 포맷 : pdf

## 1. 팀 정보

팀명	Ceres	팀장	이창준
팀원	배노일	팀원	차순웅
팀원	이예진	팀원	김귀중한

## 2. 개발완료보고서

### 0. 작품명

#### 1. 개요

##### 1.1. 작품 개요

특수바퀴(나사 추진 바퀴 내부에 Folding Wheel을 내재)를 이용한 비 평탄 지형 주행이 가능한 농업용 로봇

##### 1.2. 필요성

- 현 한국 농업사회의 인구감소 및 고령화  
추후 대한민국 농업 인구의 감소 가속화와 더불어 농촌 인구의 고령화에 따른 대체 노동력의 제공과 보다 편리하고 쾌적한 농 작업 개선의 필요성이 대두되고 있다. 이에 환경 로봇기술을 활용한 혁신적인 첨단 농업 환경을 조성하여 농업 경쟁력을 확보할 필요성이 있다. 첨단 농업용 로봇을 사용하면 소수의 인력으로 넓은 경작지를 재배할 수 있어서 농업 분야의 생산성과 경제성을 증가시킬 것이다.
- 농업용 로봇의 문제점  
해외에서도 농업용 로봇기술 개발에 대한 관심은 집중되고 있으며 다양한 농업용 로봇이 출시되었다. 하지만 이는 해외 농업환경에 맞춘 로봇이며 한국형 발농사 로봇 기술개발의 필요성이 대두된다. 따라서 저탄소 녹색성장과 친환경 트렌드에 맞추어 인간공학적인 기술을 통해 환경을 보전하면서 수확량을 줄이지 않고 고품질 농산물을 생산의 로봇기술 개발이 필수적이다.
- 나사 추진 바퀴와 접이식 바퀴를 접목한 농업용 로봇  
최근 다양한 농업 로봇이 일반 캐터필러나 바퀴를 사용한 로봇이 개발되었지만 밭고랑에서의 이동 또는 특수한 환경에서의 이동이 한계가 있는 것으로 나타났다. 이에 본 팀은 나사 추진 바퀴 내부에 접이식 바퀴를 내재하여 전 방향 이동이 가능한 특수 바퀴를 설계하여 논, 밭, 산 등 다양한 환경에서도 주행이 가능한 농업용 로봇을 개발하였다. 또한, 다관절 로봇 팔을 사용하여 파종, 수확 등의 전체적인 농 작업을 수행할 수 있으며 이와 같은 로봇 개발을 통하여 농업 자동화 시스템을 구현해서 소수의 인력으로 넓은 경작지 재배를 가증하게 할 것으로 예상된다.

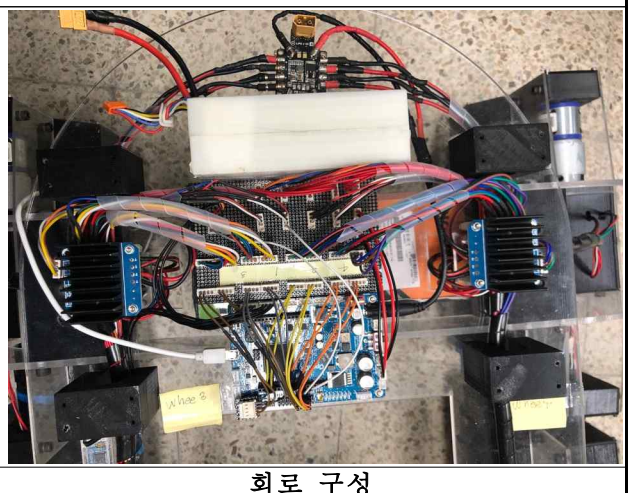
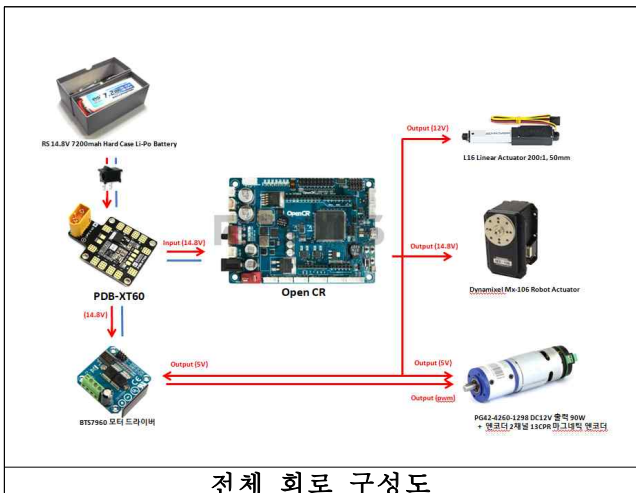
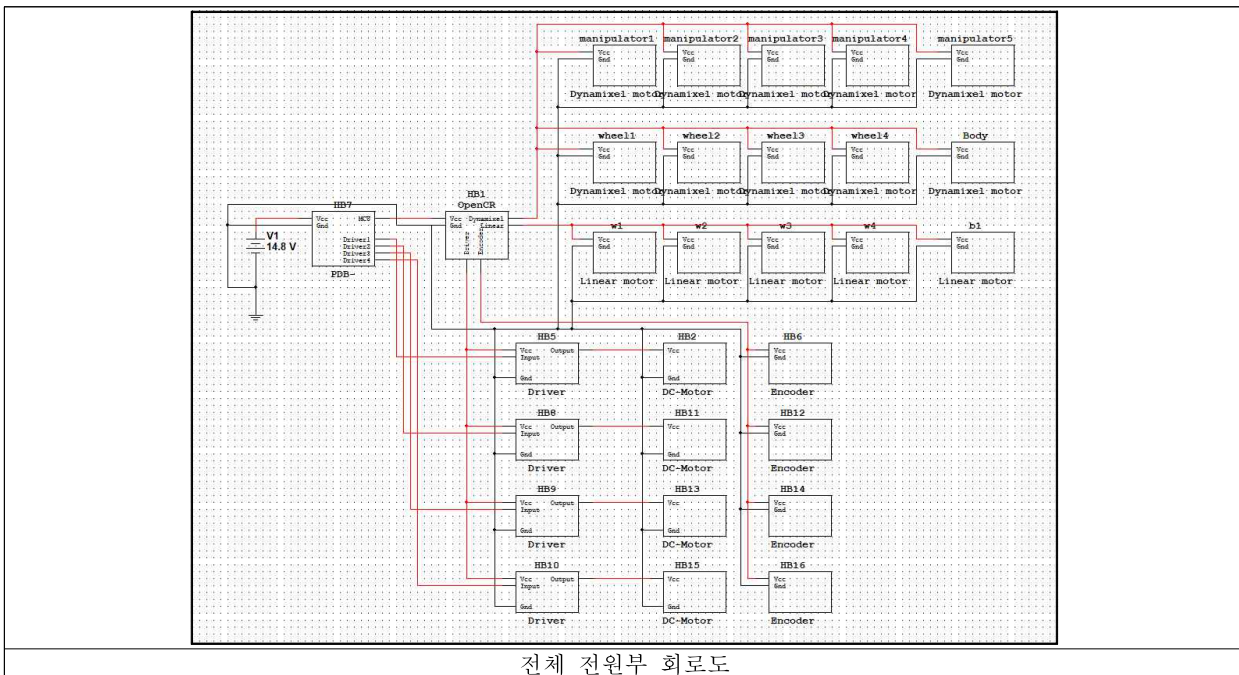
### 1.3. 개발 목표

기존의 농업용 로봇은 방향 전환이 어려우며 장애물 회피에 어려움이 있었다. 또한 도로의 환경에 제약을 받으며 로봇의 속도 및 기동성에 취약하여 효과적이라고 보기에 어려움이 있었다. 이러한 로봇의 이동 문제점을 바탕으로 본 T.F.B팀에서는 농업용 로봇을 개발하기 위해 다음의 목표를 설정하고 개발을 진행한다.

- 밭의 환경을 고려하여 흙길에서 기동성을 확보하기 위해 나선형 형태의 바퀴인 'Screw Wheel'을 제작한다.
- 일반 평지에서 Screw Wheel의 기동성을 보완하기 위해 내부에 'Folding Wheel'을 접목시켜 도로 환경에 구애받지 않도록 한다.
- 작물 파종을 위해 로봇의 몸체에 땅을 파는 크레인 역할을 할 수 있는 기능을 추가.
- 물체를 옮기기 위한 5DOF Manipulator를 접목시켜 작물을 옮겨 심을 수 있는 기능 구현.
- Depth Camera를 이용하여 전방의 도로 상태(밭고랑 또는 일반 평지)를 인식하고 바퀴의 구동 방식을 자율적으로 바꿀 수 있도록 한다.

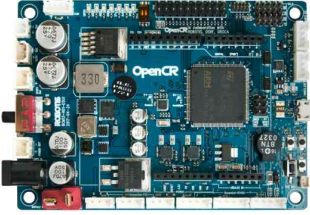
## 2. 작품 설명 (최대한 자세하게 기술)

### 2.1. HW 구성




## 2.1-1 작품 자원


### 1. OpenCR(Controller)

	Microcontroller	STM32F746ZGT6/32-bit ARM Cortex®-M7 FPU (216MHz, 462DMIPS)
	Programmer	ARM Cortex 10pin JTAG/SWD connector USB Device Firmware Upgrade (DFU) Serial
	Communication circuits	USB (Micro-B USB connector/USB 2.0/Host/Peripheral/OTG) TTL (B3B-EH-A / Dynamixel) RS485 (B4B-EH-A / Dynamixel) UART x 2 (20010WS-04) CAN (20010WS-04)
	Powers	5 V (USB VBUS), 7-24 V (Battery or SMPS)
[그림 2-1] OpenCR		


### 2. Odroid(Embedded Board)

	Processor	CortecTM-A15 2Ghz CortexTM-A7 Octa core CPU
	Memory	2Gbyte LPDDR RAM PoP stacked
	Communication circuits	2 x USB 3.0 Host, 1 x USB 2.0 Host
[그림 2-2] Odroid		


### 3. Motor Driver

	Input Voltage	DC 6 ~ 27V
	Control Voltage	3.3 ~ 5V
[그림 2-2] BTS7960	Maximum Current	43A


### 4. Intel Real Sense(Depth Camera)

	Maximum Video Resolution	1920 x 1080
	Maximum Frame Rate	30fps
	Host Interface	USB 3.0
[그림2-3] Depth Camera		


### 5. Odroid Camera

	Input Voltage	USB BUS Power(5V/280mA)
	Interface	USB 3.0 Super-Speed
	Pixel size	1.4 μm x 1.4 μm
	Lens	Standard M12 Lens with focal length of 3.6mm
[그림2-4] Odroid Ocam		


### 6. DC Motor

	Input Voltage	DC 12V
	Max Current	22.45A
	Max Output Power	191.16W
	Max Torque	325.61mNm
	Gear Ratio	24:1
[그림2-5] DC Motor		


### 7. 엔코더

	Input Voltage	DC 5V
	Channel	2 Channel
	Gear	13CPR
[그림2-5] 엔코더		


### 8. Linear DC Motor

	Input Voltage	6V
	Gear Ratio	210:1
	Peak Power Point	45N @ 2.5mm/s
	Peak Efficiency Point	18N @ 4mm/s
[그림2-6] Linear Motor		


### 9. Dynamixel DC Motor

	Input Voltage	14.8V
	MCU	Cortex-M3 (72 [Mhz], 32 [bit])
	Gear Ratio	225 : 1
	Link (Physical)	RS485 Multi Drop Bus
	Stall Torque [N·m]	8.40
	Baud Rate	8,000 ~ 4,500,000
	Operating Mode / Angle	Wheel Mode : Endless turn
		Joint Mode : 360 [deg] Multi-turn Mode : ±28 [rev]
[그림2-7] MX-106R		

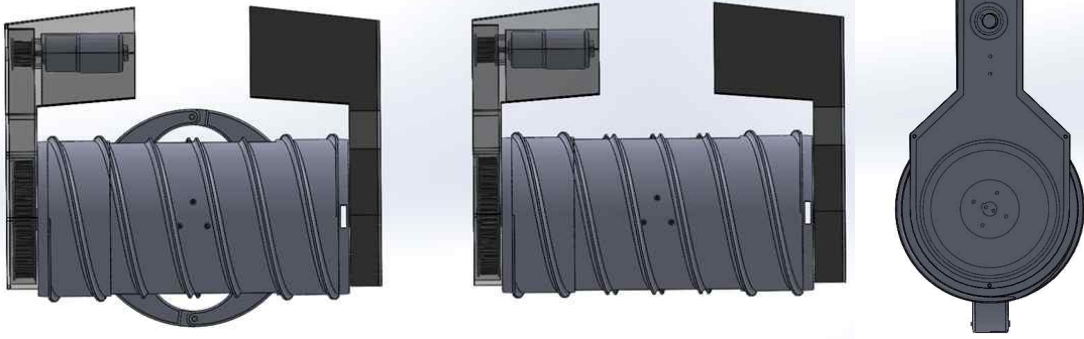
### 10. 슬립링

	Max A	2 A
	Max V	240 V
[그림2-5] 슬립링	Wires	6 wires

### 11. 타이밍벨트

	Size	670mm
	규격	STPD형 S5M/5mm
[그림2-5] 타이밍벨트		

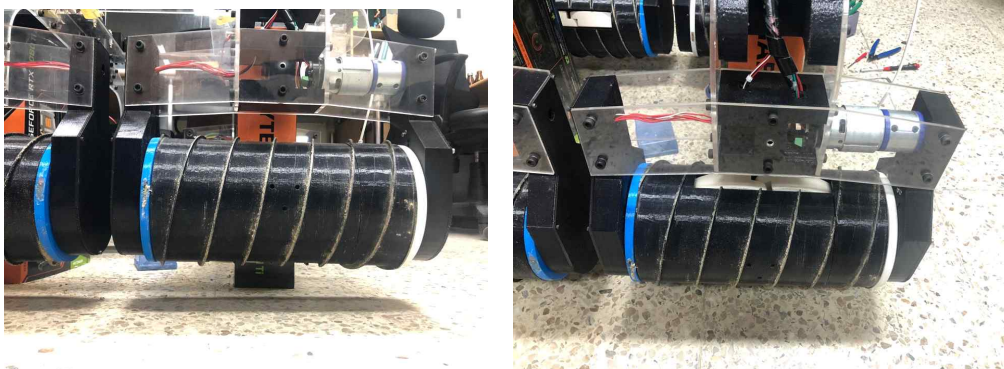
## 2.1-2 Screw Line Wheel



Screw Line Wheel 설계도면



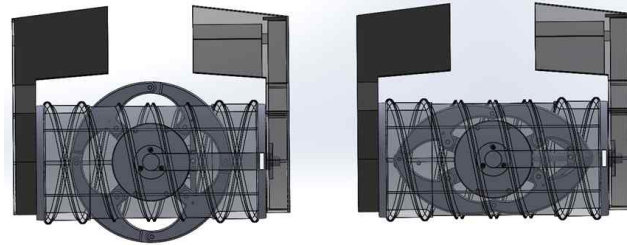
Screw Line Wheel



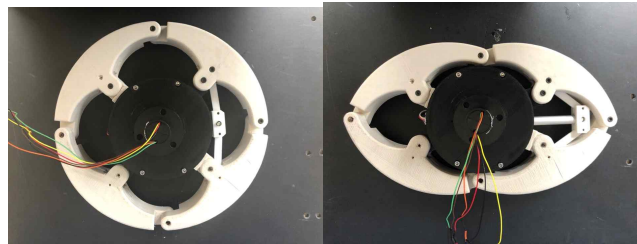
Screw Line Wheel

1. Screw Line Wheel의 소재를 3D Print를 이용하여 경량화 함
2. Screw Wheel은 DC Motor를 이용하여 구동
  - 엔코더를 이용하여 위치를 제어할 수 있도록 설계
3. Screw Line Wheel을 고정하는 부분
  - Slip ring을 이용하여 data를 전송 및 무한회전이 가능하게 설계
  - 기어를 설계하여 DC Motor와 타이밍 벨트를 이용하여 전체 Screw Line Wheel 구동
4. Motor Driver를 통하여 속도와 방향을 제어.

### 2.1-3 Folding Wheel



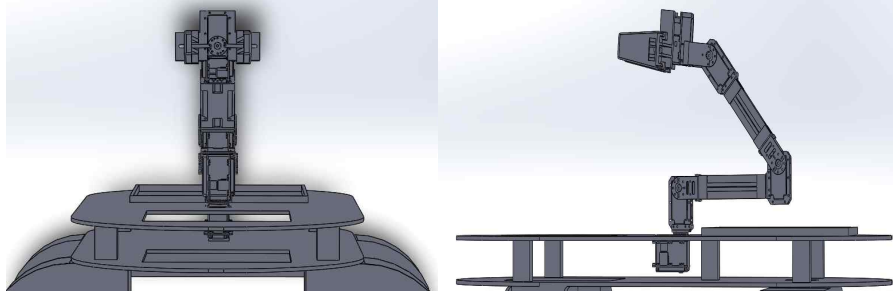
Folding Wheel 설계도면



Folding Wheel

1. Folding Wheel의 소재를 3D Print를 이용하여 경량화 함
2. Dynamixel DC Motor를 이용하여 바퀴형태의 Folding Wheel 구동
3. 내부의 Linear DC Motor를 이용하여 바퀴를 접고 펴는 기능 구현함
  - 전체 지름의 1/3의 길이를 줄이도록 설계
  - Folding Wheel 회전 시 무한 회전을 위하여 Slip ring을 이용.

### 2.1-4 Manipulator



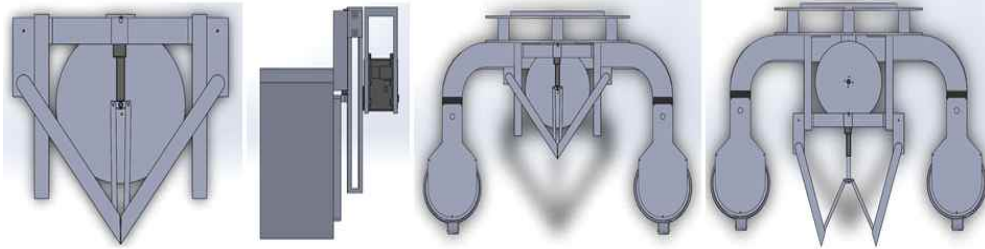
Manipulator 설계도면



Manipulator

1. 5개의 Dynamixel을 이용하여 5자유도(4개의 joint 및 1개의 gripper) Manipulator 설계
2. 소재를 3D Print를 이용하여 경량화 함

## 2.1-5 Dig Frame



Dig Frame 설계도면

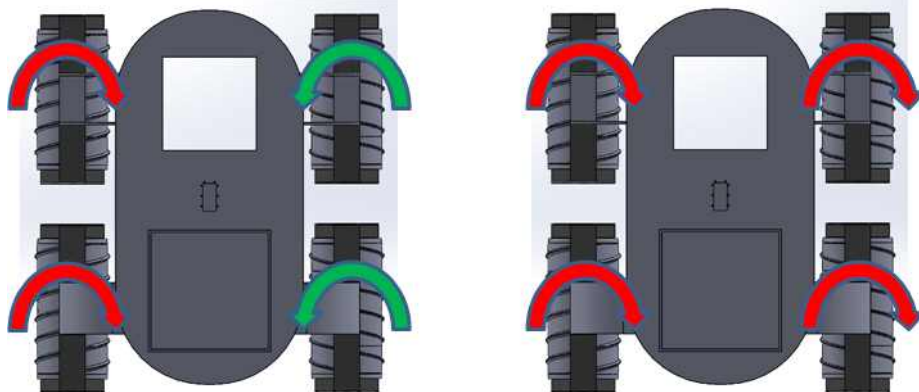


Dig Frame

1. 작물 파종 시 공간 확보를 위한 프레임 설계
2. 소재를 3D Print를 이용하여 경량화 함
3. Dynamixel DC Motor와 회전운동을 직선운동으로 변환해주는 프레임을 이용하여 상, 하 이동
4. Linear DC Motor를 이용하여 파종을 위한 프레임 동작

## 2.2. HW 기능(제어 방법 등 서술)

### 2.2-1 Screw Line Wheel



[전, 후 방향]

[좌, 우 방향]

Screw Line Wheel 주행 방식

Screw Line Wheel의 주행 방법으로는 다음의 그림과 같이 나선형태의 바퀴 4개를 다른 방향으로 회전하여 전, 후 및 좌, 우 이동을 하게 된다.

Screw Line Wheel의 제어 방법으로는 일반 주행 시 PWM 신호와 방향을 Driver로 보내주어 DC Motor를 이용하여 주행하며, Screw Wheel과 Folding Wheel을 변환하여 사용하기 위해 Screw Line Wheel의 PWM 신호를 위치 P제어를 이용하여 중량을 맞춘다. 이때 Screw Line Wheel의 중량을 맞추는 방법으로는 DC모터와 엔코더의 기어비를 통하여 312펄스가 출력되었을 때 모터가 한 바퀴 회전하고 Screw Line Wheel을 제작하며 설계한 기어가 4.5 : 1로 총 1404번의 펄스가 측정되었을 때 Screw Line Wheel은 한 바퀴를 회전하게 된다. 즉 702번의 펄스가 나올 때 방향에 상관없이 Screw Line Wheel의 중량을 맞추고 Folding Wheel을 사용할 수 있는 상태가 된다.

## 2.2-2 Folding Wheel



Folding Wheel 주행

Folding Wheel의 주행 방법으로는 병진 속도 제어를 통하여 주행하며 전진 및 후진 좌, 우 회전이 가능하다.

Folding Wheel에서 Screw Wheel을 변환하여 사용하기 위해서는 내부로 들어가기 위한 위치로 이동해야한다. 모터의 속도 P제어를 통하여 Folding Wheel의 접히는 부분이 원하는 위치에 있도록 하고 이후 Linear DC Motor에 의해 Folding Wheel이 접히게 되고 Screw Line Wheel 내부로 완전히 들어가게 된다.

## 2.2-3 Manipulator



Manipulator 구동

Manipulator 구동 방법으로는 Dynamixel DC Motor 내부에 포함된 Potentiometer를 이용하여 Dynamixel의 현재 위치를 읽어오는 방식으로 구동을 하였다. 각각의 지정된 Position으로 이동하는 경우에는 Manipulator의 자연스러운 움직임을 위해 cos함수를 통한 sampling을 이용하여 구동하였으며, 물체의 Position으로 이동할 때는 Jacobian 수식을 통하여 Object의 위치와 현재 Manipulator의 End effect의 위치의 오차를 줄여가며 Pick & Place를 수행할 수 있도록 기능을 구현하였다.

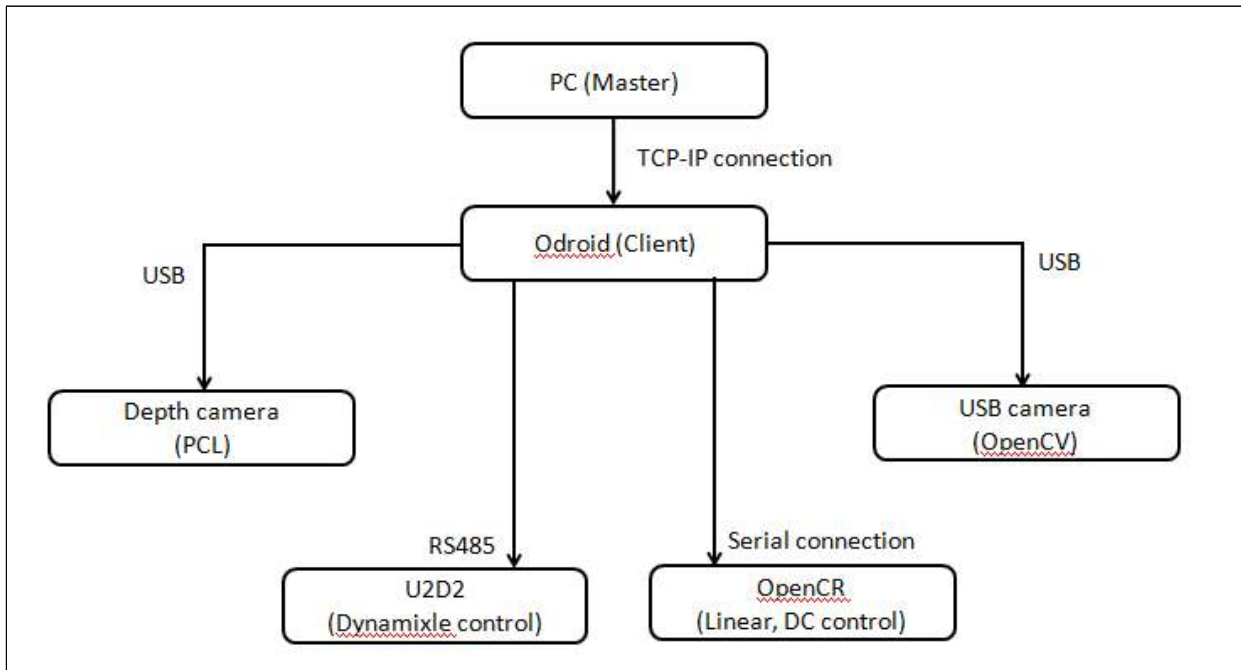
## 2.2-4 Dig Frame

모종의 파종을 위해 공간을 확보하기 위한 Dig Frame은 Dynamixel의 회전운동을 직선운동으로 변환해주도록 설계한 프레임에 장착하여 구동된다. Dynamixel의 회전운동에 의해 Dig Frame은 상, 하로 이동하며 모종삽의 기능을 하며, 공간 확보를 위해서는 Linear DC Motor를 구동하여 Dig Frame을 양 옆으로 벌려 공간을 확보한다.

## 2.3. Software 구성

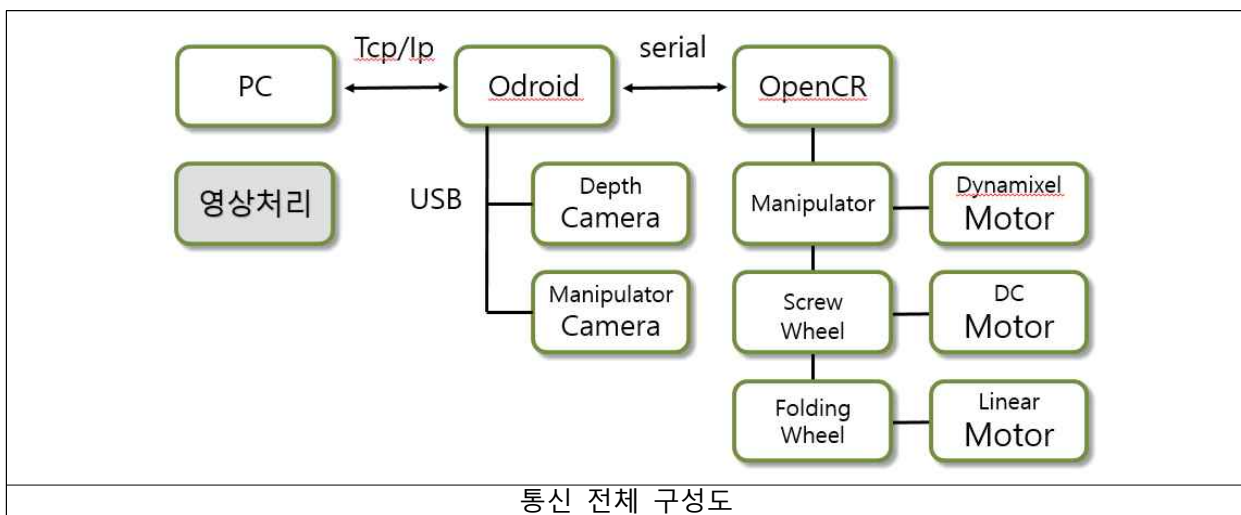
### 2.3.1 Software 구성도

#### 1. 전체 구성도



- 서버와 클라이언트의 TCP-IP통신 (같은 네트워크에 존재할 때 가능)
- Odroid (Client)에서 명령어를 받아 각각의 역할별로 CR, Node별로 message 전송하여 동작 구현
- OpenCR은 Odroid로부터 Serial통신으로 명령어를 받아 Linear와 DC 모터를 구동
- Depth camera는 ROSpackage를 통하여 message를 생성 후 PC로 보내준다.
- USB camera는 Odroid에서 OpenCV lib를 사용하여 좌표를 인식한 후 명령어에 따라 manipulator로 전송한다.

#### 2. Tcp / Ip 통신 & Serial 통신



<b>Publisher :</b> PC_control	<b>Publisher :</b> OD_control servo OD_dynamixel_wheel OD_dynamixel_body OD_dynamixel_mani Mposition	<b>Publisher :</b> cmd_vel offset mode controller_dynamixel	<b>Publisher :</b>
<b>Subscriber :</b> OD_control Depth_poitncloude	<b>Subscriber :</b> PC_control controller_dynamixel	<b>Subscriber :</b> OD_dynamixel_wheel OD_dynamixel_body OD_dynamixel_mani Mposition	<b>Subscriber :</b>
PC	Odroid	Dymanixel_control	realsense
Node 별 Topic Message ( ROS/Message 통신 )			

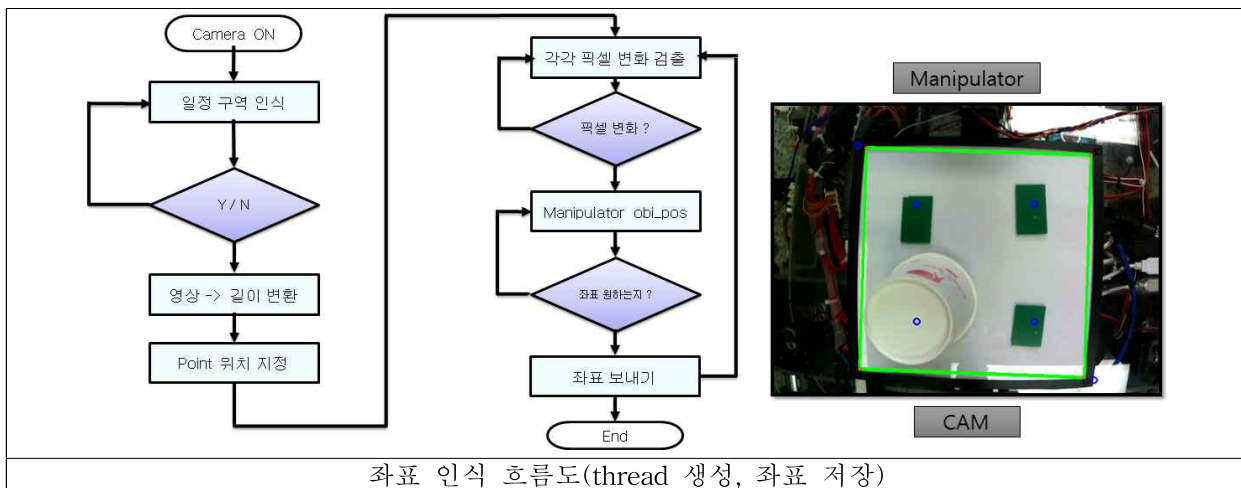
1. Mode
  - 수동(0),자동(1)
2. Param1(Wheel\_Linear)
  - 대기(0) 집기(1) 퍼기(2) 오프셋(3)
3. Param2(Wheel\_Linear)
  - 대기(0) 전진(1) 후진(2) 우향(3) 좌향(4) 우회(5) 좌회(6)
4. Param3(Wheel\_Linear)
  - 대기(0) 전진(1) 후진(2) 우향(3) 좌향(4) 오프셋(5)
5. Param4(Wheel\_Linear)
  - 대기(0) 실행(1) 종료(2)
6. Param5(Wheel\_Linear)
  - 대기(0) 하강(1) 상승(2)
7. Param6(Wheel\_Linear)
  - 대기(0) 실행(1) 종료(2)

PC - OD 명령어(& Serial)

- Odroid에서 PC로 image 전달
- PC에서 영상처리를 진행 후 명령어 전달
- serial 통신을 통한 Screw Line Wheel & Linear Motor 제어

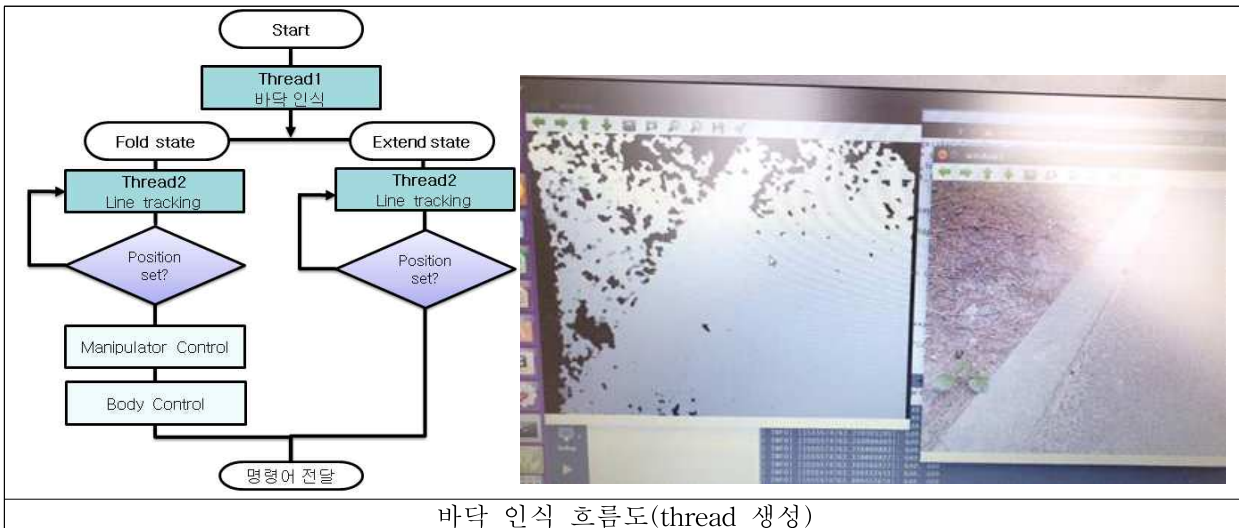
### 3. 영상처리

- USB camera(Manipulator position)



- camera에서 작업 구역이 인식되었을 때 개수와 위치를 인식.
- Manipulator좌표계와 USB camera의 좌표계를 변환
- 시작 명령어가 주어지면 좌표를 전송 및 완료 명령어 대기.

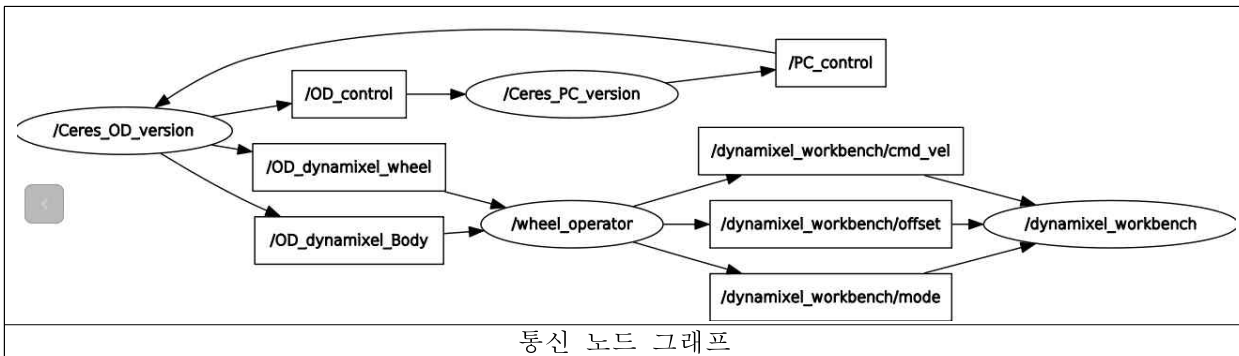
■ Depth camera (Wheel 변환을 위한 바닥 인식)



- Depth camera Point Cloud 알고리즘을 이용한 지면 인식
- 지면인식 후 주행방법 결정
- Depth camera(rgb\_image) 허프변환 알고리즘을 이용한 Line Tracking과 자세 제어

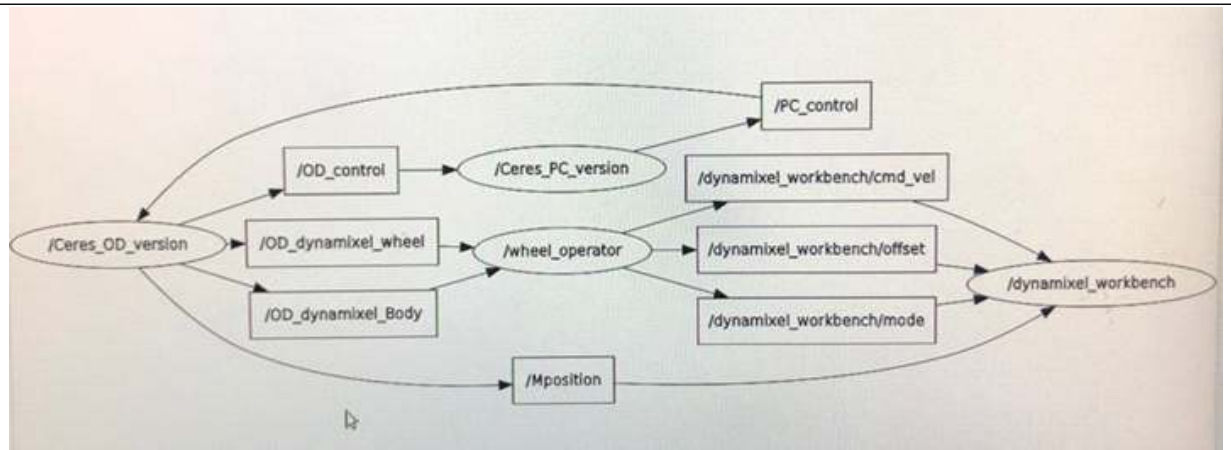
2.4. Software 설계도(흐름도 및 클래스 다이어그램 등 (개발언어에 따라 선택))

1. Tcp / Ip 통신 & Serial 통신

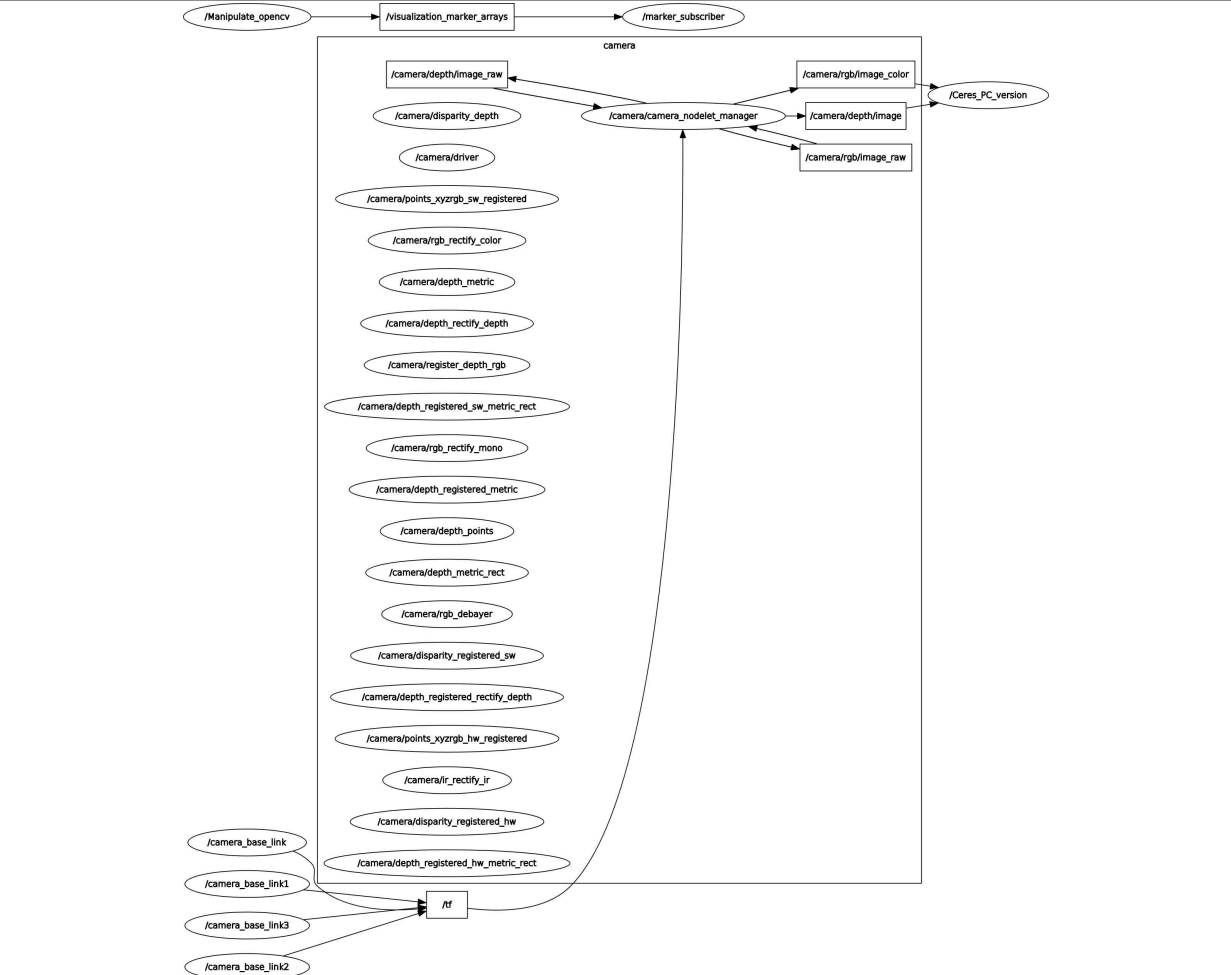


- Ceres\_PC와 Ceres\_OD는 Ceres\_msg를 통하여 서로 cmd를 주고 받는다.
- Ceres\_OD는 받은 명령어를 분별하여 명령을 수행한다.
- Dynamixel은 U2D2를 통하여 485통신을 node로 구동.
- OPenCR을 통신은 node가 아닌 Serial 직렬 통신을 이용.

## 2. 영상처리



Manipulator position( 좌표) topic 추가 그래프



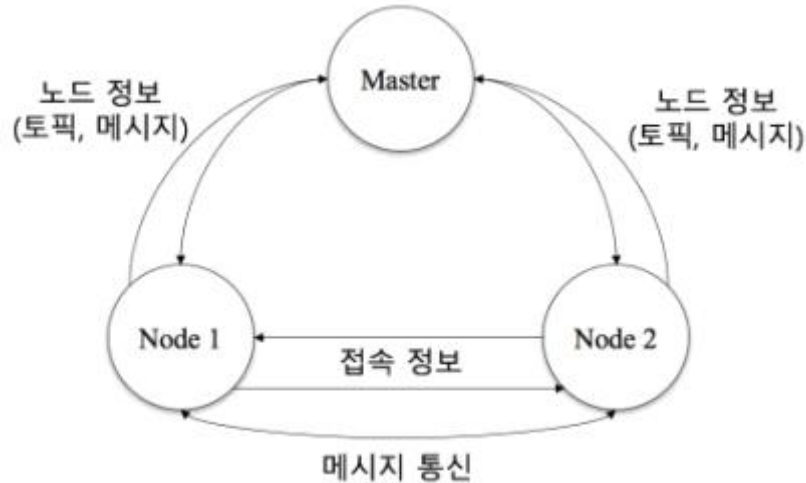
Depth camera Topic

- realsense2\_camera Package를 통하여 Topic을 전송.
- Ceres\_PC에서 받아 PCL을 사용하여 영상처리를 진행.
- 바닥인식 부분과 라인 인식을 통하여 로봇의 초기 자세 off\_set.
- USB camera를 통하여 Manipulator에 보낼 좌표를 출력

## 2.5. Software 기능 (알고리즘 설명 포함)

### 1. Tcp / Ip 통신 & Serial 통신

ROS(Robot Operation System)는 모든 실행 처리를 노드(Node)라는 개념으로 처리한다. 아래 그림은 노드 간의 통신 과정을 보여주는데 Master를 중심으로 각 노드 간의 통신을 한다. 주로 토픽이랑 메시지 통신을 통해 서로 데이터를 주고받는다.



Host PC(server)에서 Master를 실행하고 ODROID와 TCP/IP 통신을 통해 메시지를 주고받는다. ODROID는 Serial 통신을 통해 제어기인 OpenCR과 roserial을 사용해서 메시지통신으로 모터를 제어한다.

### 2. 영상처리

USB camera에서 실제 길이를 픽셀단위로 계산하여 알아낸 후 실제 좌표를 알아낼 수 있다.

길이를 측정한 후 물체의 위치를 추출한 후 그 위치에 물체가 있는지 유무를 판단하여 Array에 저장을 해 놓은 후 manipulator의 시작 신호에 맞춰 publisher가 된다.

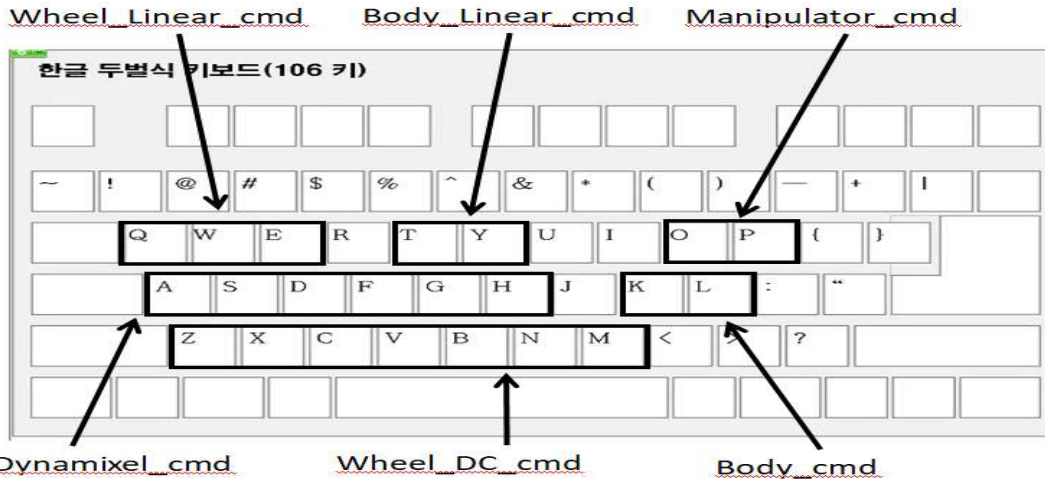


Depth camera는 IR센서를 사용하여 Point Cloud를 생성할수 있다. realsense2\_camera Node에서 Publish 하는 Point Cloud Topic을 이용하여 PCLibrary에 적용하여 바닥을 인식할 수 있다.

바닥을 인식한 후 그에 따른 명령어를 odroid로 보내주에 로봇을 구동하게 된다.

## 2.6. 프로그램 사용법 (Interface)

### ■ 수동 모드



- **Wheel\_Linear\_cmd** : Fold - Extend - Offset
- **Wheel\_Dynamixel\_cmd** : Go - Back - Left - Right - Stop - Offset
- **Wheel\_DC\_cmd** : Go - Back - Left - Right - Turn Left - Turn Right - Stop
- **Body\_linear\_cmd** : Start - End
- **Body\_cmd** : Up - Down
- **Manipulator\_cmd** : Start - End

## 2.7. 개발환경 (언어, Tool, 사용시스템 등)

ROS 는 로봇용 공개소스 메타 운영체제입니다. ROS는 일반 운영체제에서 제공하는 하드웨어 추상화, 저수준 기기 제어, 빈번히 사용되는 기능들이 구현되어 있으며, 프로세스간 메시지 전달, 패키지 관리 기능 등을 제공합니다. 또한, ROS는 여러 컴퓨터 시스템 작동하는 코드를 얻어오고, 빌드하고, 작성하고, 실행하기 위한 도구 및 라이브러리를 제공합니다.

## 3. 프로그램 설명 (최대한 자세하게 기술)

### 3.1. 파일 구성

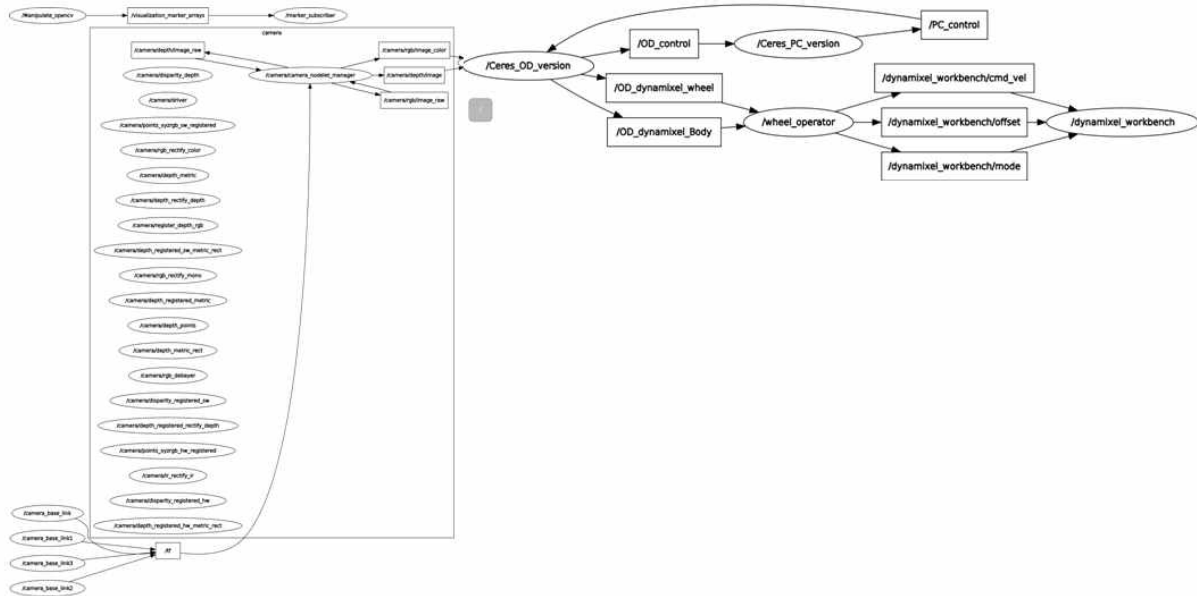
소스 파일명	소스 핵심내용
Ceres_PC (ROS master)	로봇 구동을 위한 서버 - 명령어 전달
Ceres_OD (ROS_host)	서버로부터 명령을 받아 OpenCR에 명령어 전달
RealSense2_Camera.launch	Depth 영상처리 담당
Dynamixel_control.launch	모터 셋팅, 샘플링, 동작 실행
OpenCR(Serial)	DC ,Linear Motor 구동

### 3.2. 함수별 기능

함수명	핵심기능	
Ceres_PC (ROS master)	ceres_PC:: Dirtroad_Check_depth(Mat image)	PCL이미지 사용하여 바닥 구별
	ceres_PC:: Findfurrow_Check_Depth(Mat image) Findfurrow_Check_RGB(Mat image)	RGB & Depth 이미지 사용하여 위치잡기
	ceres_PC:: LineDetect_Check_RGB(Mat image)	RGB & Depth 이미지 사용하여 라인 인식 및 주행
	Dirroad_Check() Position_Check()	이미지 Thread 함수
Ceres_OD (ROS_host)	Parameter1_info_check() Parameter2_info_check() Parameter3_info_check() Parameter4_info_check() Parameter5_info_check() Parameter6_info_check()	PC의 명령어를 분별하여 다른 명령어일때 실행.
	Check_all()	Serial보내기 위한 변수 설정
	Send_Point:: Send_pointarray	좌표 초기화
	Send_Point:: Get_pixel	좌표 유무 확인
	Send_Point:: Point_view	좌표 확인
	Send_Point:: Search_square	작업 영역 확인
	cam_read	좌표 추출하는 Thread
	Dynamixel_control.laun ch	GetObPos
ManipulatoModeSet		Manipulator 시작 Callback함수
Commend_offsetCallback		Wheel_dynamixel offset Callback 함수
commandVelocityCallback		Wheel 속도 제어 Callback함수
Manipulator::Offset		Manipulator 제어 함수
Manipulator::Move2Object		-jacobial
Manipulator::ObjectGrip		-sampling
Manipulator::Move2Bottom1		
Manipulator::MoveBottom2		
Manipulator::ObjectRelease		
Get_ODW_cmdcallback Get_ODB_cmdcallback Get_ODM_cmdcallback	ODroid 명령어 callback함수	
OpenCR(Serial)	doWheel1_decoder doWheel1_decoder doWheel1_decoder doWheel1_decoder	엔코더 값 출력
	Offset Go Back Left	PWM 펄스폭 변조 및 DC moving

	Right T_Left T_Right Stop	
	Serial_event	serial 입력 이벤트

### 3.3. 주요 함수의 흐름도



### 3.4. 기술적 차별성

□나사 추진 바퀴와 접이식 바퀴를 접목한 구동 메커니즘

현재 시중에 볼 수 있는 구동 메커니즘으로는 메카넘휠, 일반 휠, 무한궤도 등이 있다. 본 팀에서 개발한 구동 메커니즘은 시중에 볼 수 있는 구동 메커니즘과는 다르게 나사 추진 바퀴 내부에 접이식 바퀴를 접목한 형태로 일반 휠에 비교하였을 때 흙, 자갈, 눈길 등에서 나사 추진 바퀴의 우수한 성능을 보이며, 메카넘휠과 무한궤도를 접이식 휠과 비교하였을 때 일반 평지 또는 포장도로에서 더욱 우수한 성능을 보인다. 이에 나사 추진 바퀴의 장점과 일반 휠의 장점을 합하여 새로운 구동 메커니즘을 개발하였으며, 이를 이용하여 농업용 로봇뿐 만이 아닌 더욱 다양한 환경에서 구동이 가능한 모바일 로봇을 개발할 수 있을 것으로 예상된다.

#### 4. 개발 중 장애요인과 해결방안

##### 1. Screw Line Wheel 개발 과정

개발 중 장애 요인으로는 Screw Line Wheel의 설계 방법에서 나타났다. [그림 4-1]은 Screw Line Wheel의 초기 모습이다. [그림 4-1]과 같이 Screw Line Wheel을 제작할 경우 다음의 문제점이 발생하였다.

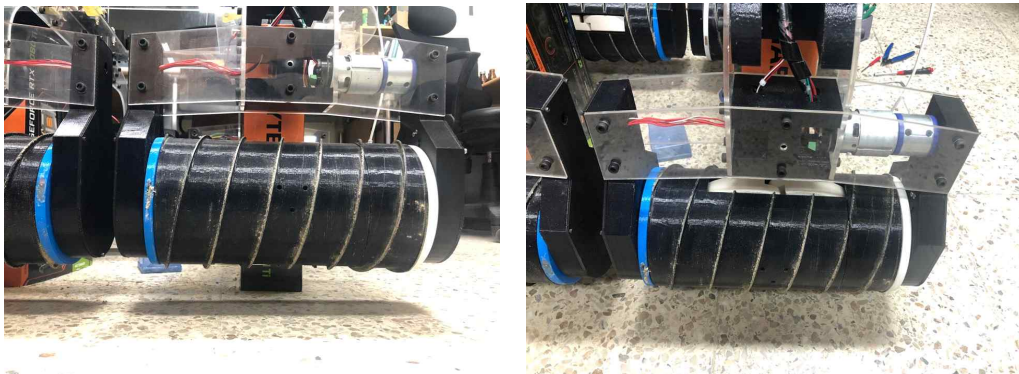


[그림4-1] 초기 Screw Line Wheel

첫 번째로 내부의 Folding Wheel의 공간을 확보하기 위해서 두 Screw Line Wheel의 사이 공간이 넓어진다는 문제점이 발생한다. 이로 인해 로봇이 주행 시 한 쪽의 Screw Line Wheel이 바닥에 접촉한 이후 반대편 쪽의 Screw Line Wheel이 바닥에 접촉하는 동시에 중간에 공간에 의해서 로봇에 큰 충격이 가해지는 것을 확인할 수 있었다.

두 번째로는 Screw Line Wheel의 형태를 타원형의 형태로 제작하여 외부에 나사선을 입힌 형태로 바퀴를 제작하여 문제가 발생하였다. 바퀴를 타원형의 형태로 제작할 경우 발생하는 문제점으로는 바닥과 접촉할 수 있는 나사선이 중간의 나사선 몇 개만 접촉한다는 것이다. 나사선 바퀴의 구동 방식은 접촉면과 나사선의 마찰로 이동하는 것인데 이 경우 바닥과 접촉하는 나사선의 개수가 매우 적다는 문제점이 발생한다.

위와 같은 문제점을 해결하기 위해 [그림 4-2]와 같이 Screw Line Wheel을 설계하였다.



[그림 4-2] Screw Line Wheel 최종 형태

초기 Screw Line Wheel에서의 첫 번째 문제점으로는 내부 Folding Wheel의 공간을 확보하기 위해 Screw Line Wheel 사이의 간격이 넓어서 생기는 문제점이 있었다. 이에 대한 해결 방법으로는 그림 2에서 볼 수 있듯이 Folding Wheel이 완전히 펼쳐져있을 경우 Folding Wheel을 구동시킬 수 있는 최소한의 공간을 제외한 모든 부분을 Screw Line Wheel로 감싸는 방법을 선택하여 해결하였다. 이 결과 두 Screw Line Wheel의 불필요한 공간의 여백을 없앨 수 있었으며, 또한 주행 중 생기는 충격을 없앨 수 있었다.

두 번째 문제인 바퀴의 형태를 타원형으로 제작하는 경우에 대해서는 바퀴의 형태를 타원형이 아닌 원통형으로 설계하여 문제를 해결할 수 있었다. 바퀴를 원통형으로 제작할 경우 바퀴 외부의 나사선

이 바닥면에 전부 접촉하여 마찰력을 최대한 사용할 수 있었으며 이는 결과적으로 바퀴의 추진력을 더욱 끌어올릴 수 있었다.

## 5. 개발결과물의 차별성

Cere팀의 결과물은 지식과 정보, 기술 융합의 시대를 맞아 10년 내 가장 유망한 사업이자 우리나라의 가장 취약한 산업인 농업의 경쟁력 확보를 위해 로봇기술 적용관점에서 조망했다. 세계농업의 로봇화를 이끄는 동인을 파악하고, 선제적 산업화 기술개발을 통한 농업과 로봇 연관 산업의 동반성장을 모색했다.

기존의 농업용 바퀴에 많이 사용되는 것은 무한궤도 또는 캐터필러 궤도형 바퀴로 2개 이상의 바퀴와 그 둘레에 두른 판을 이용한 추진 방식이다. 표면적이 넓기 때문에 타이어보다 차량의 무게를 분산시켜 약한 땅에서도 가라앉거나 갇히지 않을 수 있어 주로 사용하고 있으나 이동에 있어 자유롭지 못하고, 기동성이 낮은 것을 볼 수 있다. Screw Line Wheel은 수륙 양용 차량으로 개발되어 어려운 눈과 얼음 또는 진흙 늪에 대처하기 위한 목적으로 개발된 특수한 바퀴로 한쪽은 시계 방향으로 회전하고 다른 한쪽은 반 시계 방향으로 회전하여 회전축을 따라 전진 또는 후진하는 원리로 이동한다. 바퀴의 회전 방향을 통해 전 방향으로 이동이 가능하다.

Screw Line Wheel은 흙길 또는 자갈에서 기동성이 우수하지만 일반 평지에서는 기동성이 약하다는 단점이 있다. 이러한 단점을 보완하기 위해 내부에 타이어 형태의 바퀴를 추가하여 일반 평지에서 기동성을 확보할 수 있는 바퀴를 개발하였다. Screw Line Wheel과 Folding Wheel의 구동 방식으로는 Depth Camera를 이용하여 영상처리를 통해 전방의 지형 상태(밭고랑 또는 평지)를 인식하여 자율적으로 구동 방식의 변화를 줄 수 있도록 하였으며, 이는 결과적으로 도로 환경에 구애 받지 않도록 할 수 있다.

## 6. 단계별 개발계획 및 실제 참여인원 및 업무 분장

Cere팀은 제어공학과 2명과 기계공학과 1명으로, 로봇을 만들기 위한 최적의 조화를 이루는 팀으로 구성하였다. 기계공학적 요소를 해석 및 적용하여 로봇을 설계하고 제어를 위한 code를 작성하는 방식으로 진행하였다.

기계공학의 기계설계를 통해 배운 나사의 회전력 및 토크에 관련한 이론을 구동부에 적용하고, 이를 통해, 바퀴에 대한 마찰력 계산이나 회전방향에 대해 참고하여 진행하였다, 또한 로봇 공학의 정 기구학 및 역기구학 등을 토대로 5DOF Manipulator를 제어하는데 있어서 수월하게 개발 진행을 할 수 있었으며 기계적인 것들을 고려하여 설계된 로봇과 ROS(Robot Operating System)를 적용하여 Manipulator와 Screw Line Wheel 및 Folding Wheel의 움직임을 제어하고 서버와 클라이언트(로봇)의 통신 및 영상처리를 통하여 자율적 이동과 자연스러운 동작을 구현하였다.

지도교수님은 한경대학교 제어공학과 김용태 교수님으로 많은 제어 및 지능시스템 관련 프로젝트의 경험을 바탕으로 주 1회 이상의 회의를 통해 제작하고 있는 로봇의 필요한 요소와 고려해야 하는 것들의 대한 설명과 feedback을 받는 시간을 가졌다. 진행과정 중 로봇의 장·단점 및 문제점을 파악하여 수정이 필요한 부분의 방향과 방식을 잡아주어 개발을 진행하였다. 예를 들어, Body의 파는 부분은 회전운동을 직선운동으로 바꾸는 방식과 Screw Line Wheel을 구동하기 위해 필요한 모터의 토크와 지면과의 마찰을 고려하여 Motor를 결정할 수 있게 조언을 해주셨다.

팀원들 또한 개인 및 타 학교와의 프로젝트를 통한 경험을 바탕으로 영상처리, 모터 구동을 위한 알고리즘 등을 개발 진행 중인 로봇에 적용하였다. 팀원 모두 본 대회 및 창원 전국 지능로봇 경진대회, 공학 페스티벌 등에 참여한 경험이 있으며 꾸준히 입상하여 성적을 거두었으며, 2018년도 한국지능로봇 대회의 작품들 중 S.F.B팀으로 참가를 하였다. 당시 심사위원분들의 질문들에서 로봇의 많은 부분이 부족하다고 생각되어 수정과 변화를 주어 본 로봇 개발을 진행하게 되었고 2019년도 올해 초부터 작품을 설계하여 S.F.B의 단점을 보완하고 역할을 추가하고 로봇의 목적과 완성도를 중점으로 개발을 진행 하였다.

지능로봇 개발 경험으로는 ROS 플랫폼과 LDS를 이용한 SLAM Navigation, Manipulator제작 및 구동, 휴머노이드 로봇 구동을 위한 알고리즘 개발, Depth Camera와 USB cam 등을 이용한 물체인식 및 회피 등의 개발을 진행한 경험이 있다.