

기술의 특장점

- 측정대상 **제품 자동 선택** (QR코드 비전인식)
- 품온 측정시간을 기존 제품 대비 **최대 5배 이상 단축**
- 측정된 제품 **이미지를 포함하여 데이터 자동 전송** 및 실시간 기록
- 기존 기술 및 제품 대비 **운영 비용 70% 이상 절감**

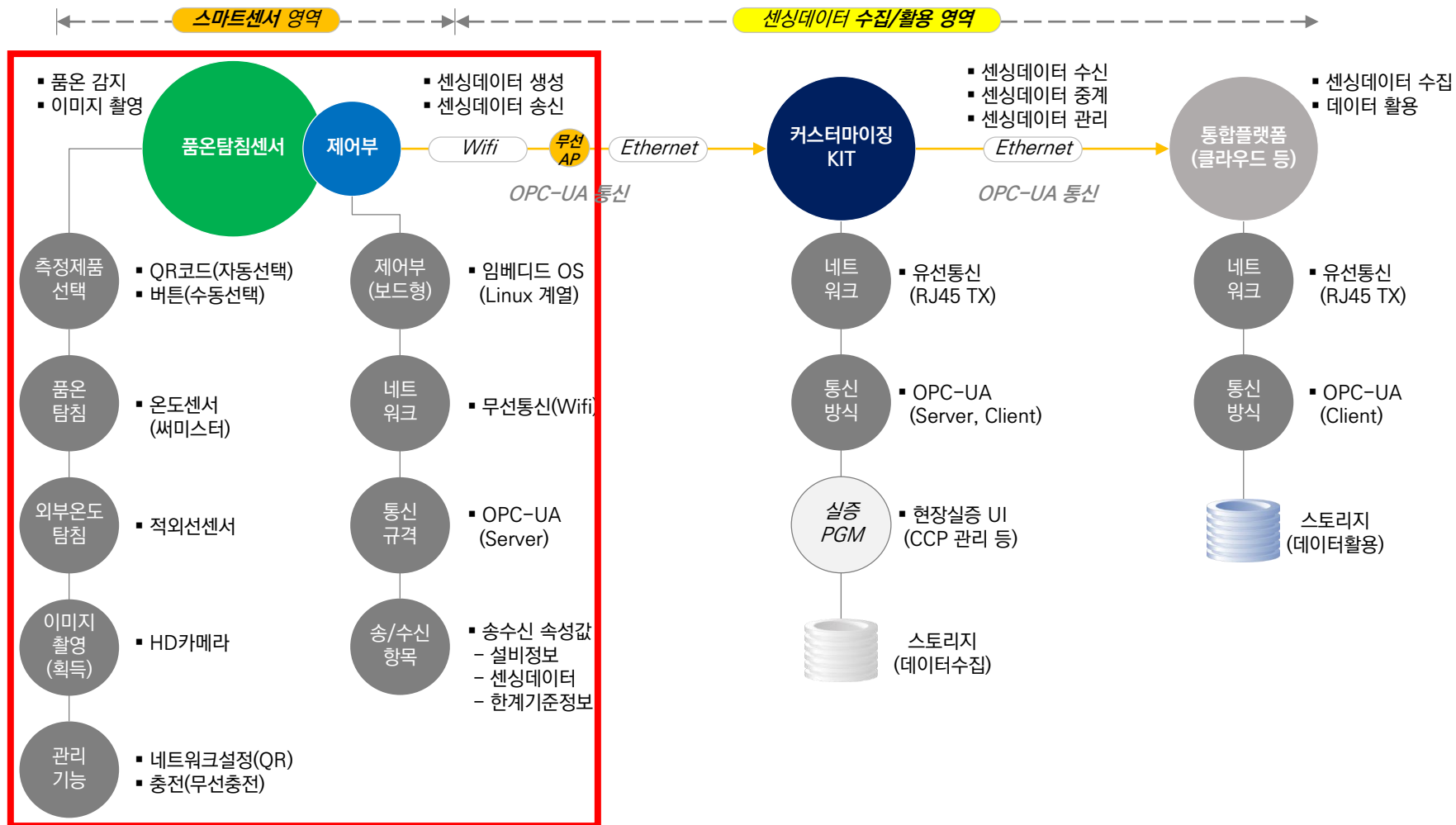
제품 구성 내역

제품 이미지 획득을 위한
HD급 카메라

제품 내/외부 온도 측정을 위한
탐온센서, 적외선센서

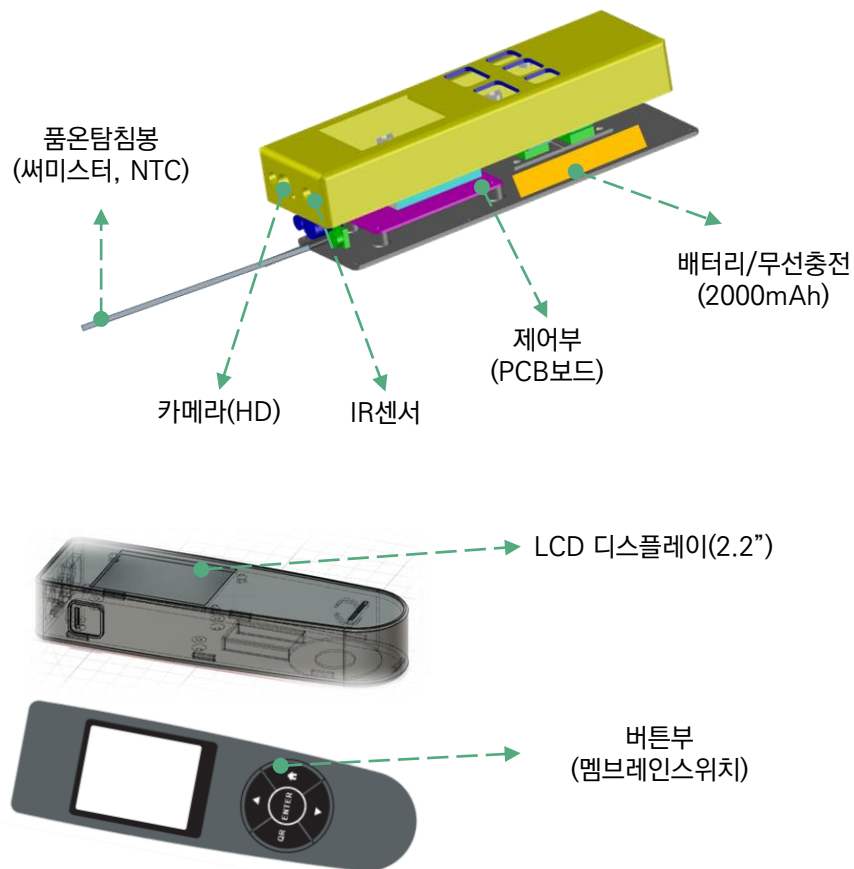
이미지 및 데이터 전송을 위한
제어부(PCB)

기능구조



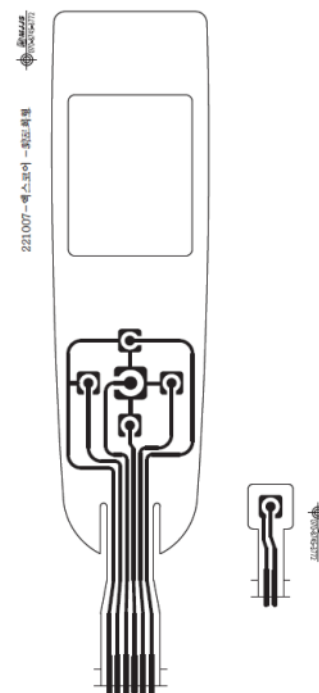
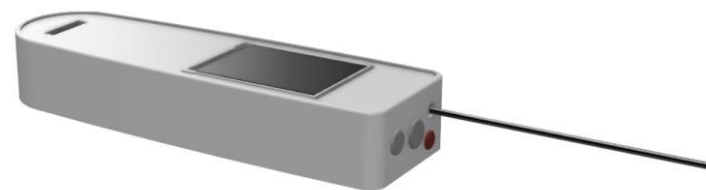
제품 구성도

- 구성 : 제품 및 충전거치대
- 크기 : 320(W) x 50(D) x 25(H)mm



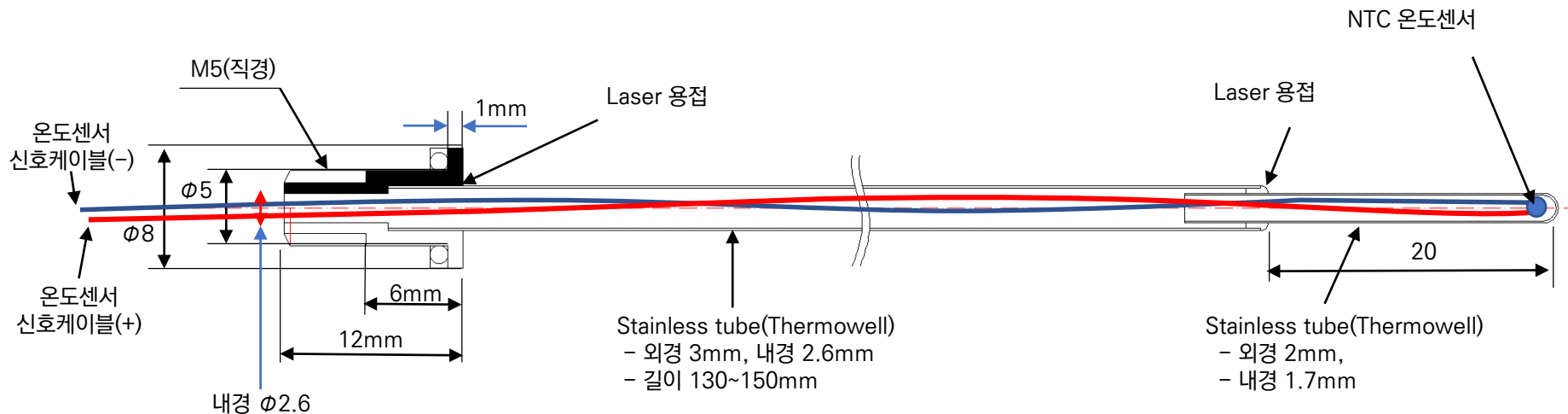
3D 설계도

※ 본체 3D 프린터용 도면 첨부



※ 멤브레인스위치 도면 첨부

탐침봉 설계도



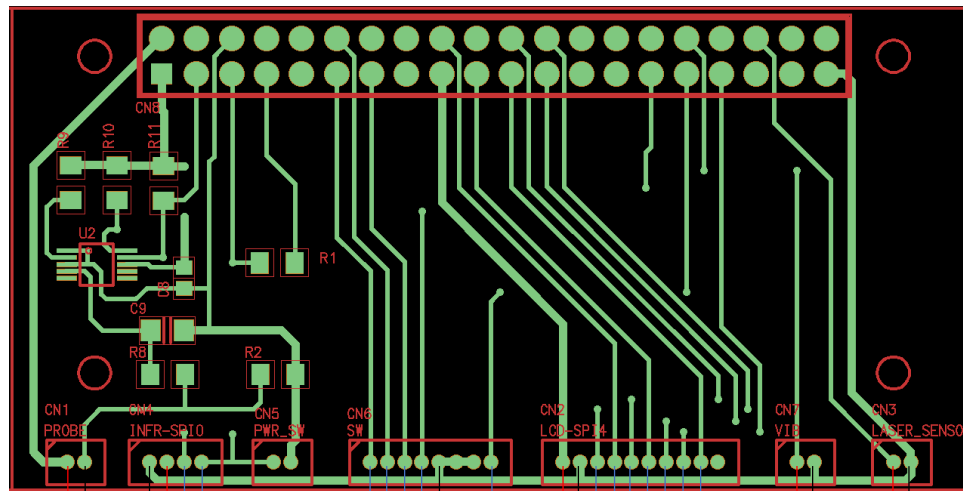
※ 탐침봉 연결부는 SUS 재질 가공 후 용접

※ 탐침봉은 스테인레스 주문 제작
 ※ 용접은 스테인레스 용접
 ※ Sensor 끝부분 레이저 용접 후 라운드 처리
 ※ 반드시 Laser 용접(2부분) 후 NTC 온도센서를 삽입 및 조립

PCB 모듈 결선도

※ 상세 설계 도면 첨부함

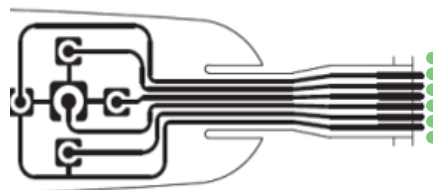
위치	부품명
R1	사용안함
R2	저항 100K
R8	저항 4.7K
R9,10,11	저항 1K
C8	캐패시터 100nf
C9	캐패시터 1uf
U2	ADC 칩



온도센서



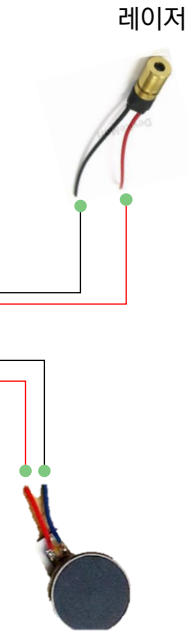
IR센서



멤브레인스위치



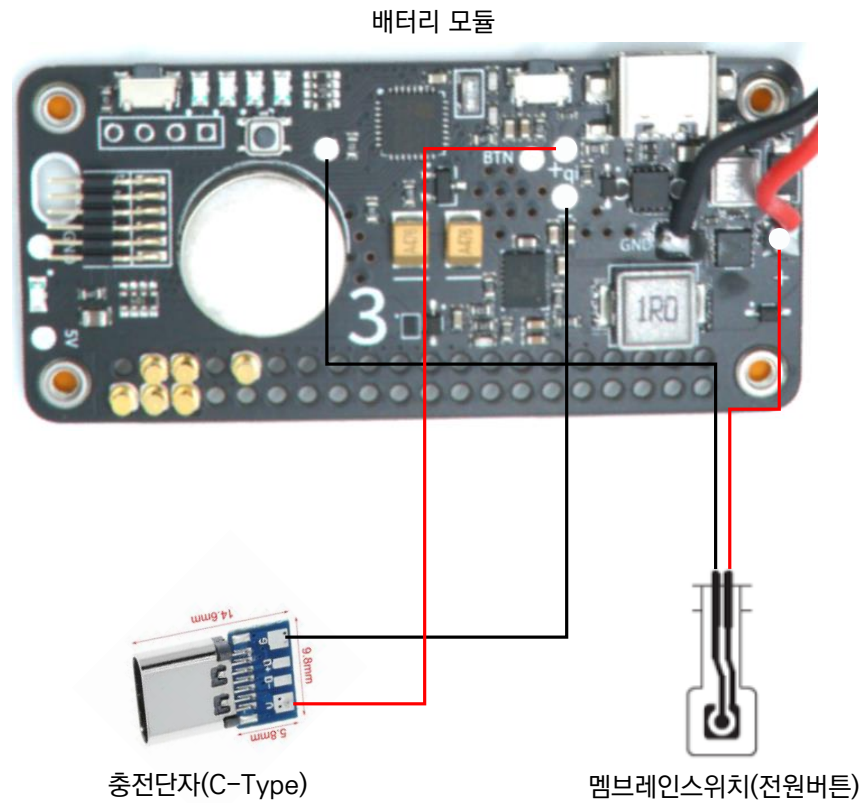
LCD



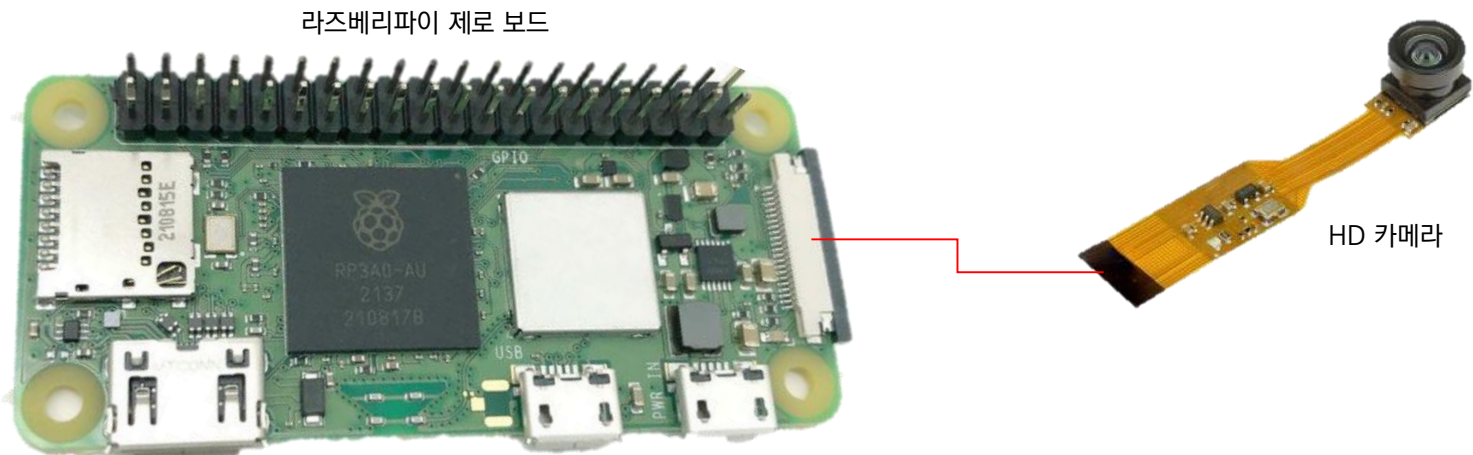
레이저

진동모터

배터리 모듈 결선도


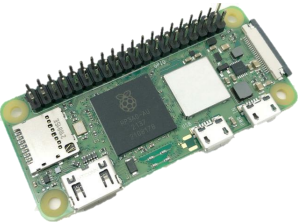
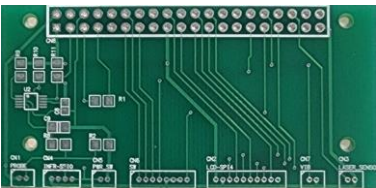


카메라 부품 결선도



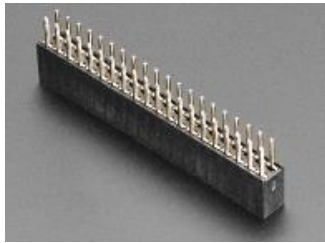
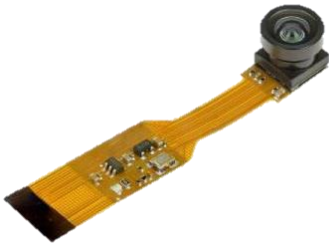
H/W 조립 준비물

제품단가 기준일 : 2023년 03월 01일

부품(시제품)	이미지	필요수량	단가	비고
TFT LCD 2.2인치		1개	20,000	모델명 : SZH-EK096 통신 : SPI 시리얼 (4 I/O) 해상도 : 320 x 240 전압 : DC 3.3V
라즈베리파이 제로 보드		1개	98,000	모델명 : 라즈베리파이 제로 2W
PCB		1개	5,000	# 주문생산 필수 - 설계도면첨부 → [도면] PCB

H/W 조립 준비물

제품단가 기준일 : 2023년 03월 01일

부품(시제품)	이미지	필요수량	단가	비고
커넥터		1개	3,000	모델명 : GPIO Header for Raspberry Pi HAT 옵션 : 2x20 짧은 암 헤더
HD 카메라		1개	35,000	모델명 : B006605 브랜드 : Arducam 렌즈 : 광각 160도 모듈 : 라즈베리파이 제로용

H/W 조립 준비물

제품단가 기준일 : 2023년 03월 01일

부품(시제품)	이미지	필요수량	단가	비고
IR센서		1개	103,000	모델명 : D6T-8L-09H 통신방식 : I2C 측정온도 : -0°C ~+ 200°C
레이저 모듈		1개	3,500	모델명 : LB laser 사이즈 : 6.5x17.5mm 전압 : 3V 전류 : 20mA 색상 : 레드(Red)
온도센서(탐침프로브 포함)		1개	1,200	모델명 : NTC 100k 3950 측정온도 : -50°C ~+ 260°C ※ Thermowell(탐침프로브)는 설계도를 참고하여 주문제작 필수 → [도면] 탐침프로브





H/W 조립 준비물

제품단가 기준일 : 2023년 03월 01일

부품(시제품)	이미지	필요수량	단가	비고
배터리 모듈		1개	52,000	모델명 : Pisugar3 용량 : 1200mA 통신방식 : I2C # 배터리 포함
멤브레인스위치		1set	50,000	전면부 : 165mm*46mm 측면부 : 16.5mm*16.5mm # 주문생산 필수 - 설계도면첨부 -> [도면] 멤브레인스위치
본체(함체)		1set	30,000	3D프린터 출력 -> 연구시 사용 모델 : Ultimaker S5 Pro 재료 : PLA 최소 충진률 : 40% # 주문생산 또는 자체생산 필수 - 설계도면 첨부 -> [도면] 품온함체

H/W 조립 준비물

제품단가 기준일 : 2023년 03월 01일

부품(시제품)	이미지	필요수량	단가	비고
C-Type 커넥터		1개	1,000	타입 : C-Type 커넥터 : 24핀 소켓 : 암
ADC 칩		1개	8,000	모델명 : ADS1115IDGSR 해상도 : 16bit 채널 수 : 4
소형 진동모터		1개	3,600	모델명 : PN-VM102 사이즈 : 10mm X 2mm 전원 : DC 3V
저항		1개	100	제품명 : 칩저항 4.7K 사이즈 : 2012 등급 : F급
		1개	100	제품명 : 칩저항 100K 사이즈 : 2012 등급 : F급
		3개	100	제품명 : 칩저항 1K 사이즈 : 2012 등급 : F급

H/W 조립 준비물

제품단가 기준일 : 2023년 03월 01일

부품(시제품)	이미지	필요수량	단가	비고
캐패시터		1개	100	모델명 : C3216 용량 : 1uf
		1개	100	모델명 : C3216 용량 : 100nf
케이블		색상별 1식	10,000	제품명 : 배선용 와이어 모델 : UL-AWG30
볼트		길이별 5개	300	제품명 : 유두렌치볼트(STS) 두께 : M2.5 길이 : 4mm, 15mm
SD 카드		1개	7,000	제조사 : SanDisk 제품명 : MicroSD Class10 ULTRA A1 용량 : 32GB

H/W 조립 순서

Step 1.

조립 준비

전체부품 준비

자체제작 및
주문제작품 구매



함체 준비

자체제작(3D프린팅)
및 주문제작품 구매

Step 2.

준비된 부품 조립

탐침온도계 조립

탐침봉 파이프에
NTC 온도센서를 삽입 후 고정

제어부보드에
부품 연결(조립)

HD카메라 연결

PCB보드 부속품 조립 및 부품 연결(조립)

PCB 부속품 조립

제작한 PCB에 저항,
캐패시터, ADC칩을 납땜

PCB에 부품연결

탐침온도계, IR센서, 레이저
멤브레인스위치, 디스플레이

배터리보드 부품연결(조립)

배터리, 무선충전코일 또는 C-Type 잭
※ 충전방식에 맞게 부품 연결

Step 3.

보드 결합 및 시험

보드 결합 (총 3개)

제어부 (라즈베리파이 제로)
PCB (부속품 통합 보드)
배터리 (배터리 관리보드)

전원 기동 및 시험

전원인가 (무선충전 시 충전 후 진행)
부품별 기능 시험 및 통합 성능 시험

※ 외장부품 연결 및 고정은 환경에 맞도록 다양한 방법으로 진행 가능

S/W 구동 및 접속 방법

제품단가 기준일 : 2023년 03월 01일

1. 디지털 품온탐침센서 전용 OS를 부팅이 가능하도록 준비

- 첨부된 OS 이미지를 micro-SD카드에 OS 형식으로 굽는다.
- OS 파일명 : smarthaccp_iot1_20230306.iso
 - ※ 모든 필수 설정과 필수 라이브러리가 설치되어 있는 OS 이미지 파일
- 첨부된 rufus-3.21p.exe 파일을 이용하여 준비한 micro-SD카드에 첨부된 OS를 설치한다.
 - ※ rufus-3.21p.exe 파일은 OS 이미지를 micro-SD로 쉽게 설치해주는 무료 프로그램

2. 무선 네트워크 설정

- 설치가 완료된 micro-SD카드에서 아래 파일을 무선 환경에 맞게 수정해준다.
- wpa_supplicant.conf

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
network={
    ssid="접속하고자 하는 wifi 이름"
    psk="접속하고자 하는 wifi 비밀번호"
    key_mgmt=WPA-PSK }
```

3. 설치가 완료된 micro-SD카드를 라즈베리파이 제로 보드에 결합 후 전원버튼을 눌러 시스템을 구동한다.

4. 시스템 접속

- 무선 공유기에서 접속된 디지털 품온탐침센서 IP를 확인한다.
 - ※ 보통 내부 네트워크 정보에서 단말기명이 "raspberrypi"를 찾으면 된다.
- ssh 접속 : port 22 / ID haccp / PW haccp

구분	구현 기술	활용 영역
운영체제	• Raspbian OS	Firmware 기반 변경 활용 가능
개발언어	• Python v3	가장 경량화된 HW 개발 언어
활용 Library	<ul style="list-style-type: none"> • 라즈베리파이 Library <ul style="list-style-type: none"> - digitalio - board - busio - picamera - Rpi.GPIO - gpiozero • 부품 제조사 Library <ul style="list-style-type: none"> - LCD 모듈 (adafruit_rgb_display) - ADC 컨버터모듈 (adafruit_ads1x15) - 배터리중계 모듈 (pisugar-power-manager) - IR센서 모듈 (d6t-8lh) 	운영체제별 오픈소스 Library 활용 가능

주요 환경 설정

1. 고정 IP 설정

- dhcpd.conf 파일 수정
\$ sudo nano -i /etc/dhcpd.conf
- 고정 IP와 ROUTERS 정보 수정

```
# Example static IP configuration:
#interface wlan0
static ip_address=192.168.0.22
#static ip6_address=fd51:42f8:caae:d92e::ff/64
static routers=192.168.0.1
#static domain_name_servers=192.168.0.1 8.8.8.8 fd51:42f8:caae:d92e::1
```

2. 데이터 전송을 위한 환경 설정 파일 수정

- thermometer.py 파일 수정
\$ nano /home/haccp/thermometer.py
- 호출 url 및 parameter 값을 해당 서버 정보에 맞게 수정

```
param['mesure_date'] = transDate
param['mesure_ymd'] = dates[0]
param['mesure_time'] = dates[1]

param['lcns_no'] = '19760015001'
param['limit_item_cd'] = 'CH006'
param['ccp_type_cd'] = 'C0010'

param['fermentation_no'] = selectedItemSub[1]
param['tank_no'] = selectedItemSub[2]
param['tank_nm'] = selectedItemSub[3]
param['tank_limit'] = selectedItemSub[4]
param['infra_id'] = selectedItemSub[8]

rsps = sendServer('https://shm.haccp.or.kr:5083/shai/rest/process/start', param)
```


충전 방법



디지털 품온탐침센서 밑 부분에 C타입 전원 케이블을 연결하면 충전이 됩니다.

※ 어댑터 1.2A 사용 권장
(2A 이상 사용 시 발열이 발생하고 기기고장의 원인이 될 수 있습니다.)


전원 ON/OFF 방법



- 전원 OFF 상태 : 측면 버튼을 1번 누르면 약 30초 후 정면 LCD 화면에 메뉴가 나타납니다.
- 전원 ON 상태 : 측면 버튼을 3초 이상 누르면 LCD 화면이 디지털 품온탐침센서 전원이 꺼집니다.

메뉴 이동 및 제품 선택 방법

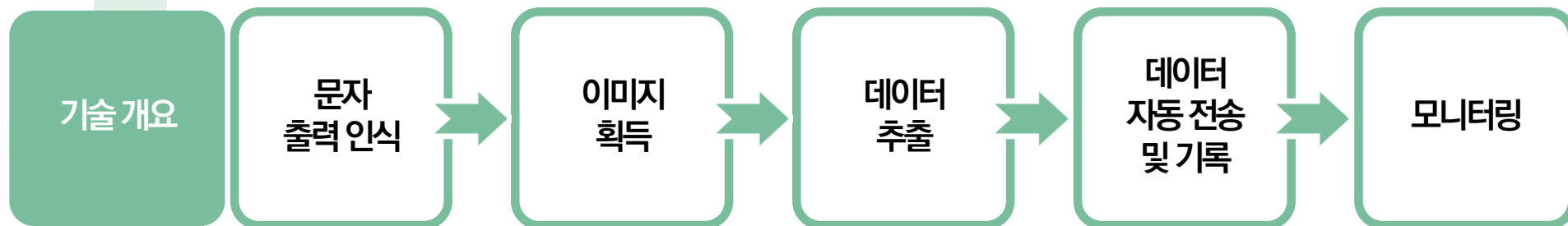


1. 오른쪽  (HOME) 버튼을 누르면 첫 화면으로 이동합니다.
2. 왼쪽 QR 버튼 또는 'QR코드 인식' 메뉴를 선택하면 QR코드를 인식할 수 있는 카메라 화면으로 이동합니다.
3. ▲(위), ▼(아래), ENTER(선택) 버튼을 통해서 화면을 선택하고 이동할 수 있습니다.
4. '제품 선택' 메뉴를 선택하면 제품 목록이 나오는 화면으로 이동하고 제품을 선택할 수 있습니다.
※ 제품 목록은 /home/haccp/thermometer.db 파일을 통해서 수정할 수 있습니다.
5. 제품을 선택하면 측정화면으로 이동합니다.
6. '네트워크 확인' 메뉴를 선택하면 현재 연결된 네트워크 IP와 배터리 잔량을 확인 할 수 있습니다.

탐침 온도 측정 방법



1. 선택한 제품과 한계기준 정보가 화면 표시되고 현재 온도와 실시간 카메라 영상이 보여집니다.
2. 온도 측정을 시작하면 레이저 포인터가 작동하여 현재 측정 중인 제품을 표시하게 됩니다.
3. 최저 또는 최고 온도까지 자동으로 측정하고 일정 시간 동안 온도변화가 없으면 측정이 종료되고 완료 팝업이 나타납니다.
4. '재측정' 선택 시 다시 측정화면으로 이동합니다.
5. '전송' 선택 시 설정한 서버로 제품정보, 현재 시간, 현재온도, 측정완료 시점의 제품 사진이 전송됩니다.
※ 전송할 서버 정보는 /home/haccp/thermometer.py 파일을 통해서 수정할 수 있습니다.
6. 전송이 완료되거나 🏠 (HOME) 버튼을 눌러 첫 화면으로 이동 할 수 있습니다.



제품 이미지

기술의
특장점

- 기존 식품 설비의 개조가 필요 없어 **설비의 유지보수와 무관하게 누구나 쉽게 설치 가능**
- 소형화 및 경량화를 통한 **설치시 공간적 제약 최소화**
- 기존 기술 및 제품 대비 구축, 운영 비용 **90% 이상 절감**

제품
구성 내역

제품 이미지 획득을 위한
HD급 카메라

이미지 추출 및 판독을 위한
AI 비전센서 모듈

구현기술

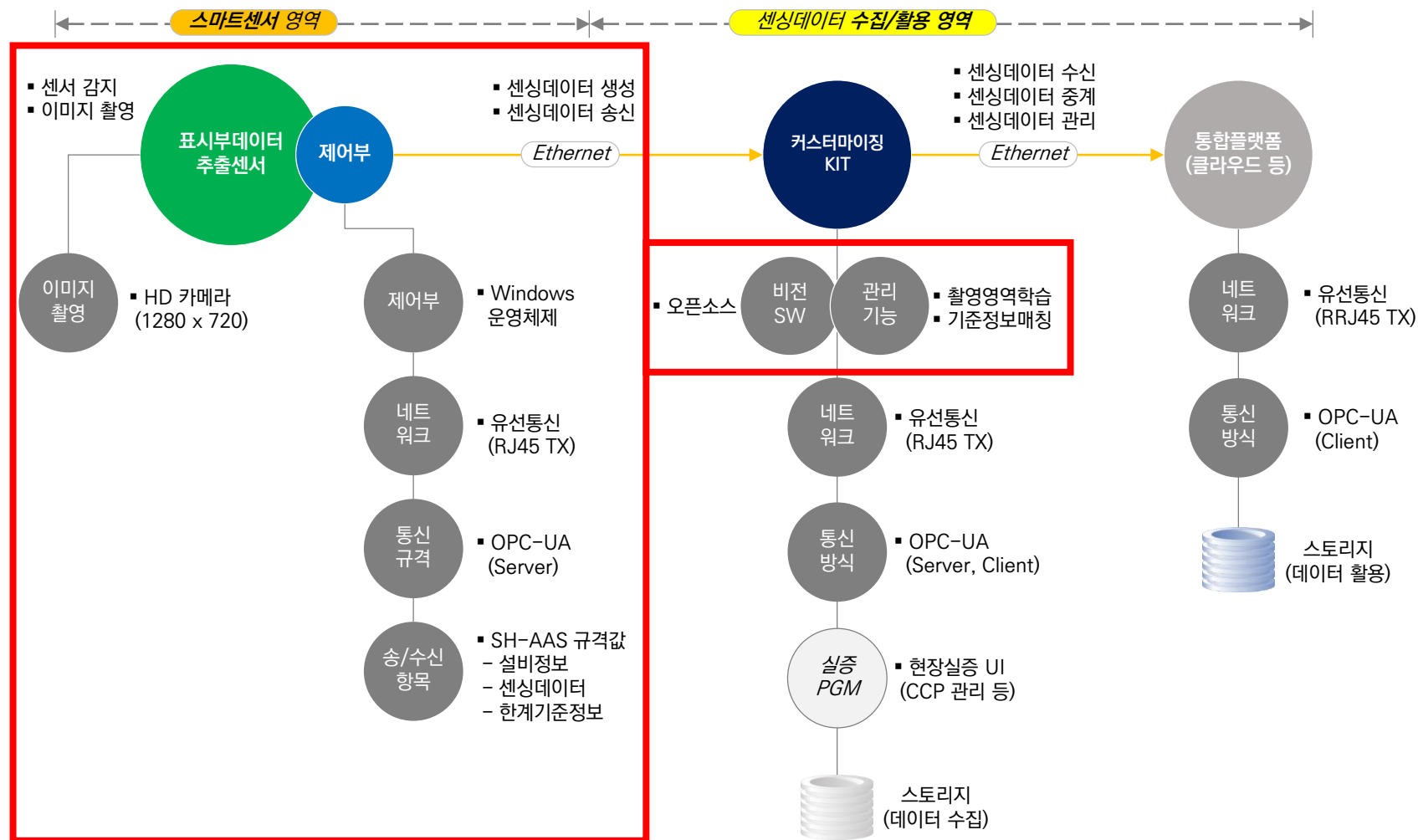
구분	오브젝트 디텍션	판독 및 전송	설비변경 최소화
구현 기능	<ul style="list-style-type: none">디지털 이미지 및 비디오 내에서 유의미한 특정 객체를 감지	<ul style="list-style-type: none">부정확한 문자 인식전처리 등 정확한 판독	<ul style="list-style-type: none">공간제약, 설비떨림현장 작업공정 방해 최소화
개발 요소	AI 비전 인식	획득이미지 판독 기술	설비별 커스터마이징 가능한 장착 기구부
	오픈소스 기술 → 제약없는 활용		



✓ 설비의 표시부 표출 내역을 중요관리점 항목으로 모니터링하는 공정 설비 샘플링

※ 외산설비, 노후설비 등 기술 지원의 제약으로 설비 개조 및 변경이 불가한 장비

기능구조



02 표시부데이터추출센서 ▶ 설계도면

제품 구성도

- 구성 : 카메라, 부착 기구부, 산업용 PC
- 크기 : 45(W) x 50(D)

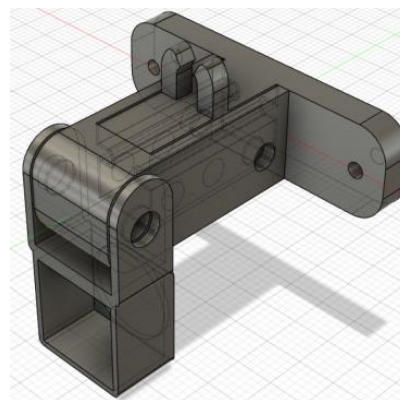
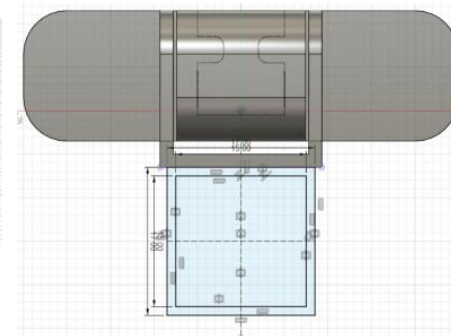
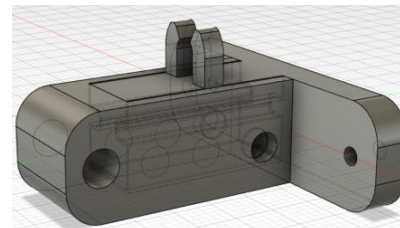


카메라 함체



3D 설계도

※ 본체 3D 프린터용 도면 첨부



H/W 조립 준비물

제품단가 기준일 : 2023년 03월 01일

부품(시제품)	이미지	필요수량	단가	비고
HD 카메라		1개	89,500	모델명 : 5MP HD USB 카메라 브랜드 : geniuspy
산업용 PC		1개	900,000	모델명 : HDL-T150PC-10CP CPU : i5 10210U Quad Core HDD : 120GB SSD 메모리 8GB DDR RAM 화면크기 : 15인치LED패널 터치방식 : 정전식 터치
본체(함체)		1set	10,000	3D프린터 출력 → 연구시 사용 모델 : Ultimaker S5 Pro 재료 : PLA 최소 충전률 : 40% # 주문생산 또는 자체생산 필수 - 설계도면 첨부 → [도면] 표시부함체

이미지 학습 방법

1. Yolov4를 이용하여 숫자 이미지 판독을 하기 위해 Google Colab를 이용하여 이미지를 학습시킨다.
Google Colab에서 Google Drive를 연결해준다.

```
[ ] from google.colab import drive
    drive.mount('/content/drive')
```

2. 작업을 진행할 폴더를 생성해주고 gitHub에서 darknet을 복사한다.

```
[ ] !mkdir yolov4
    %cd yolov4
    !git clone https://github.com/AlexeyAB/darknet
```

3. Darknet을 컴파일 하기 전 옵션을 설정해주고 컴파일을 진행한다.

```
[ ] %cd darknet/
    !sed -i 's/OPENCV=0/OPENCV=1/' Makefile
    !sed -i 's/GPU=0/GPU=1/' Makefile
    !sed -i 's/CUDNN=0/CUDNN=1/' Makefile
    !sed -i 's/CUDNN_HALF=0/CUDNN_HALF=1/' Makefile
    !sed -i 's/LIBSO=0/LIBSO=1/' Makefile
```

```
[ ] !make
```

이미지 학습 방법

4. 학습할 이미지를 오픈소스 라벨링 툴인 CVAT을 이용하여 라벨링을 진행하면 좌표값이 적힌 txt파일을 얻을 수 있다.
 학습할 이미지와 txt 파일을 압축하여 Google Drive로 업로드하고 압축을 해제한다.



```
[ ] !unzip /content/drive/MyDrive/RESEARCH/binggrae/obj.zip -d /content/yolov4/darknet/data/obj
```

5. 데이터학습에 필요한 설정파일들을 작성하고 업로드해준다.

```
[ ] !cp /content/drive/MyDrive/RESEARCH/binggrae/obj.data /content/yolov4/darknet/data
!cp /content/drive/MyDrive/RESEARCH/binggrae/obj.names /content/yolov4/darknet/data
!cp /content/drive/MyDrive/RESEARCH/binggrae/yolov4-tiny-custom.cfg /content/yolov4/darknet/cfg
```

obj.data X

```
1 classes = 14
2 train = data/train.txt
3 valid = data/valid.txt
4 test = data/test.txt
5 names = data/obj.names
6 backup = backup/
```

obj.names X

```
1 classic_fe1.5
2 classic_nonfe2.5
3 classic_sus2.0
4 classic_sus2.5
5 fe_1.5mm
6 non-fe_2.5mm
7 plain_fe1.5
8 plain_nonfe2.5
9 plain_sus2.0
10 plain_sus2.5
```

yolov4-tiny-custom.cfg X

```
1 [net]
2 # Testing
3 #batch=1
4 #subdivisions=1
5 # Training
6 batch=64
7 subdivisions=16
8 width=416
9 height=416
10 channels=3
```

이미지 학습 방법

6. gitHub에서 YOLOv4 사전학습모델을 다운로드 한다.

```
[ ] #yolov4-tiny
!wget https://github.com/AlexeyAB/darknet/releases/download/darknet_yolo_v4_pre/yolov4-tiny.conv.29
```

7. 업로드한 data, cfg, weight 파일 경로를 지정하고 학습을 진행한다.

```
[ ] #yolov4-tiny
!./darknet detector train data/obj.data cfg/yolov4-tiny-custom.cfg yolov4-tiny.conv.29 -dont_show
```

8. 학습이 완료된 모델을 이용하여 학습시킨 이미지 또는 판독할 숫자 이미지를 업로드하고 간단한 python 코드를 실행하여 테스트를 진행해본다.

```
[ ] !./darknet detector test data/obj.data cfg/yolov4-tiny-custom.cfg yolov4-tiny-best.weights test.jpg
from IPython.display import Image
Image('result.jpg')
```

S/W 구동 방법

1. 첨부된 python-3.9.exe 파일을 설치한다.

※ python 3.9 버전으로 개발되어 다른 버전일 경우 수정이 필요할 수 있습니다.

2. Python 라이브러리를 별도로 모두 설치해주거나 첨부된 라이브러리 압축파일을 Python이 설치된 경로에 Lib 폴더 안에 복사해준다.

※ python 설치 경로 예시

C:\Users\사용자\AppData\Local\Programs\Python\Python39

3. 첨부된 표시부데이터추출센서 소스코드(x_ai_ocr.zip) 압축 해제 후 config 폴더에 이미지 학습을 통해 얻은 cfg파일과 weights 파일을 복사해준다.

4. Python 프로그램 툴을 이용하여 실행하거나 Windows cmd 프로그램에서 x_ai_ocr_web.py 파일을 실행한다.

```
C:\Users>python C:\Users\x_ai_ocr\x_ai_ocr_web.py
camera-0 start
```

```
* Serving Flask app 'x_ai_ocr_web'
```

```
* Debug mode: on
```

```
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
```

```
* Running on all addresses (0.0.0.0)
```

```
* Running on http://127.0.0.1:5555
```

```
Press CTRL+C to quit
```

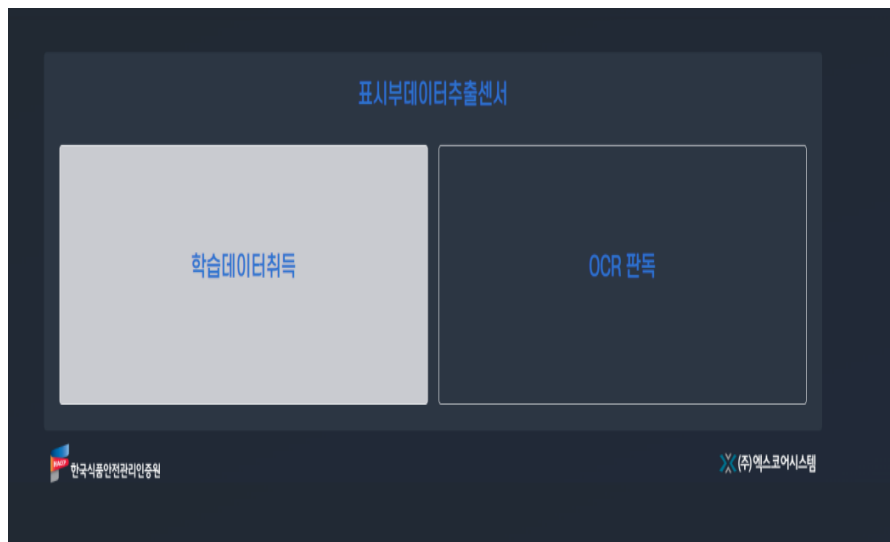
```
* Restarting with stat
```

5. 인터넷창에서 프로그램에 접속한다.

접속주소 : <http://127.0.0.1:5555> 또는 <http://localhost:5555>

구분	구현 기술	활용 영역
운영체제	• Windows 10	(권장 스택) OS 기반 운영체제 활용 권장
개발언어	• Python v3	(권장 스택) 가장 경량화된 개발 언어
비전	• YOLOv4-Tiny	오픈소스 기술 활용 (사전학습, 자동판별)
Library 활용	<ul style="list-style-type: none"> • Python pip 라이브러리 모듈 <ul style="list-style-type: none"> - flask - flask_socketio - opencv-python - numpy - flask_cors - pandas - threading - base64 - sqlite3 - requests 	운영체제별 오픈소스 Library 활용 가능

메뉴 선택 방법



1. 프로그램이 실행되면 '학습데이터취득', 'OCR 판독' 2개의 메뉴가 보여집니다.
2. '학습데이터취득' 메뉴를 선택하면 학습에 필요한 이미지 데이터를 수집할 수 있는 메뉴로 이동합니다.
3. 'OCR 판독' 메뉴를 선택하면 학습한 이미지를 기준으로 숫자를 판독할 수 있는 메뉴로 이동합니다.

학습데이터 취득 메뉴



1. 판독하려는 숫자 이미지의 인식율을 높이기 위해 영상 처리된 이미지를 학습 이미지로 생성합니다.
2. 추출 최소값, 최대값을 조정해서 판독하려는 숫자의 색상, 밝기, 주변 환경 등의 요소에도 숫자가 잘 인식될 수 있도록 값을 조정하고 저장합니다.
3. 여러 대의 카메라를 사용하는 경우 '카메라 변경' 버튼을 눌러 각 카메라 환경에 맞는 데이터를 추출할 수 있습니다.
4. 데이터 수집주기를 조정하고 '데이터 수집' 버튼을 누르면 x_ai_ocr 폴더 내에 영상 처리된 이미지가 저장됩니다.

OCR 판독 화면

가열공정

제품선택 제품명 시작 전제모드 새로고침

제품	-	제품	-	제품	-
한계기준온도	-	한계기준온도	-	한계기준온도	-
현재온도	-	현재온도	-	현재온도	-
한계기준시간	-	한계기준시간	-	한계기준시간	-
유지시간	-	유지시간	-	유지시간	-
품온기온도	-	품온기온도	-	품온기온도	-
공정상태	대기중	공정상태	대기중	공정상태	대기중

상세보기 상세보기 상세보기

추가 공정 종료 추가 공정 종료 추가 공정 종료

한국식품안전관리인증원 (주)엑스코시스템

1. 제품 선택 후 '시작' 버튼을 누르면 선택한 제품의 한계기준 온도, 시간 등의 정보가 추가되고 1개의 공정이 진행됩니다.

※ 제품 목록은 x_ai_ocr 폴더 내에 x_ai_ocr.db 파일을 통해서 수정할 수 있습니다.

2. 각각의 공정의 현재온도, 유지시간과 공정상태가 실시간으로 표시됩니다. 온도와 시간이 한계기준을 초과하는 경우 자동으로 공정상태가 이탈상태로 변경됩니다.

3. 각각의 공정을 종료, 재시작 할 수 있고 상세보기를 통해서 누적 데이터를 확인할 수 있습니다.

OCR 판독 상세 화면

가열공정

제품선택 제품명 시작 메인 전제모드 새로고침

제품	-	현재온도	-
한계기준온도	-	유지시간	-
한계기준시간	-	공정상태	대기중
품온기온도	-	공정상태	대기중

추가 공정 종료

공정번호 제품 시작시간 종료시간 유지시간 이탈여부 온도 도달시간 등록시간

974	c	2023-03-08 12:12:09	2023-03-08 12:12:11	0초	이탈	0	0초	2023-03-08 12:12:11
972	a	2023-03-08 12:10:42	2023-03-08 12:10:55	0초	이탈			
969	a	2023-03-08 12:09:35	2023-03-08 12:09:50	0초	이탈			
968	h	2023-03-08 12:07:58	2023-03-08 12:08:01	0초	이탈			
965	g	2023-03-08 12:06:17	2023-03-08 12:06:49	0초	이탈			

이전 1 2 3 4 5 34 다음

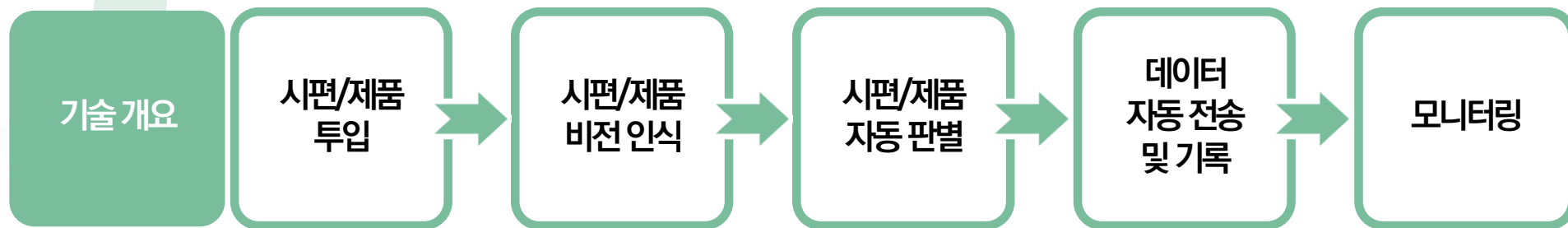
한국식품안전관리인증원 (주)엑스코시스템

1. 선택한 공정의 제품 및 한계기준 정보가 나타나고 오른쪽에 현재 판독되고 있는 카메라 영상이 실시간으로 보여집니다.

2. 왼쪽 아래 목록에 해당 오븐기의 공정 이력이 나타납니다. 가장 최근의 이력 순으로 보여지고 진행중인 공정은 맨 위에 표시됩니다. 공정 이력을 선택하면 오른쪽 목록에 해당 공정의 판독 데이터가 표시됩니다.

3. 오른쪽 아래 목록에 공정이 시작된 이후의 판독 데이터가 1초 단위로 등록됩니다.

4. 각각의 공정을 종료, 재시작 할 수 있고 상세보기를 통해서 누적 데이터를 확인할 수 있습니다.



기술의 특장점

- 시편테스트 시 **작업자 개입(투입순서, 버튼입력 등) 업무를 자동화**
- 수기로 이루어지던 **시편 테스트 과정이 자동화됨**으로써 정확한 판별 가능
- 기존 설비 변경없이, **탈부착 방식의 손쉬운 도입** 가능
- 기존 기술 및 제품 대비 구축, 운영 비용 **90% 이상 절감**

제품 구성 내역

제품 이미지 획득을 위한
HD급 카메라

이미지 획득 및 판별을 위한
AI 비전센서 모듈

구현 기술

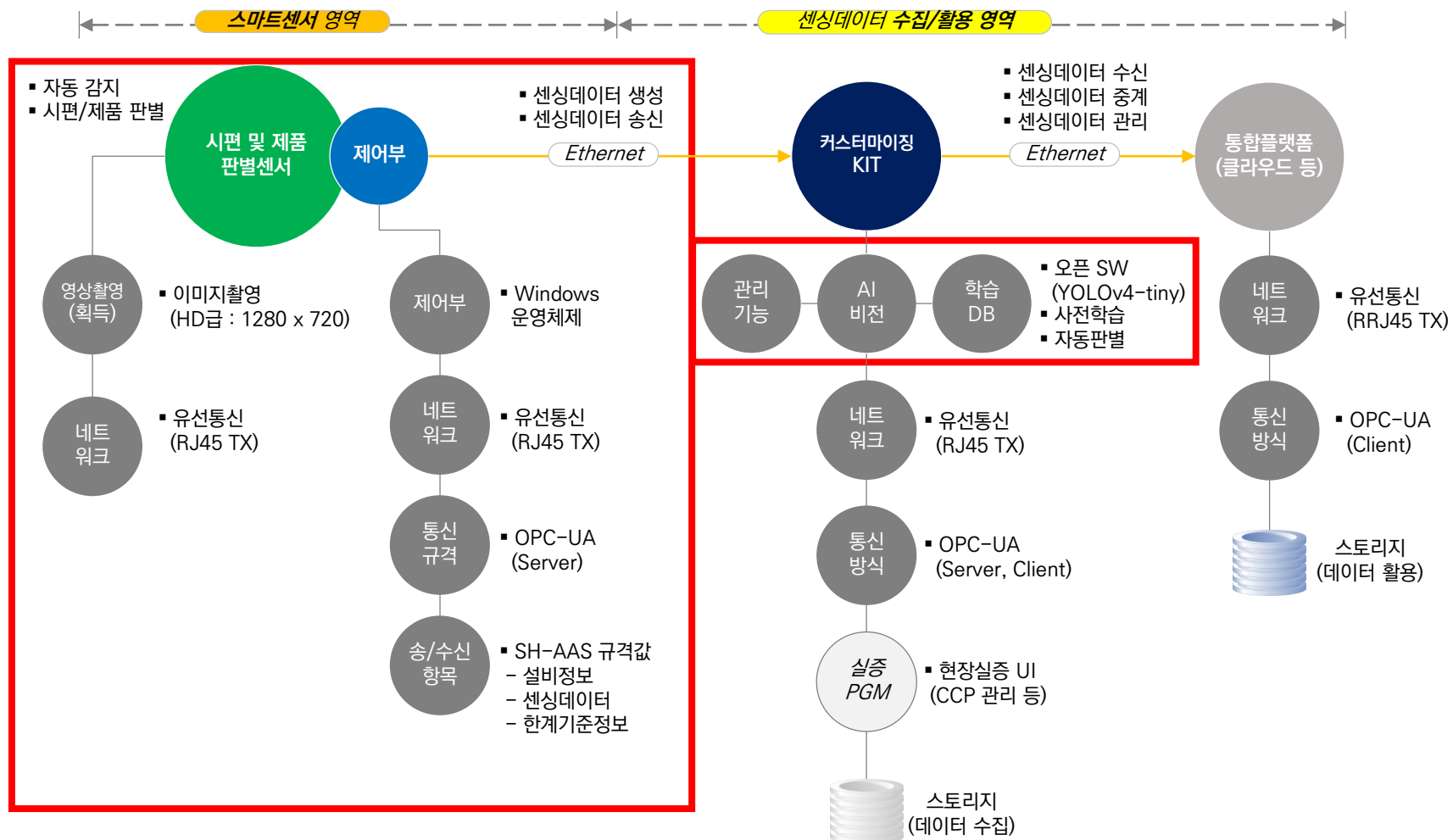
구분	오브젝트 디텍션	판별 및 전송	설비변경 최소화
구현 기능	• 디지털 이미지 및 비디오 내에서 유의미한 특정 객체를 감지	• 시편 및 제품 판별 • 사전학습 기능	• 기존설비 변경 최소화 • 현장 작업공정 방해 최소화
개발 요소	AI 비전 인식	판별 기술, 자동카운팅	설비별 커스터마이징 가능한 장착 기구부
	오픈소스 기술 지향 → 제약없는 활용		



- ✓ 산업용 PC 기반의 AI비전 운영 환경 구성
- ✓ 금속검출기와 데이터 인터페이스 연동

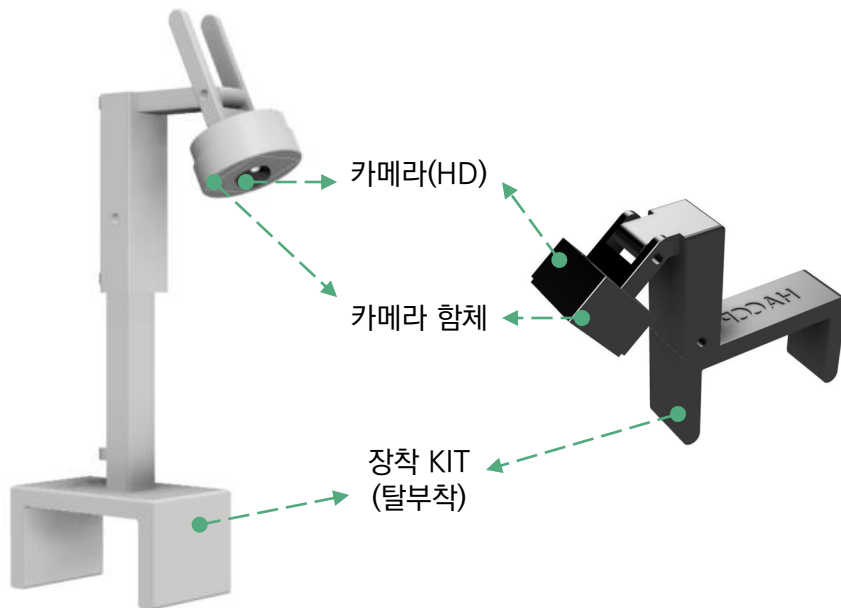
※ 중소 식품제조공장에서 가장 보편적으로 활용되는 금속검출기 설비를 대상으로 해당 기술 연구개발 진행

기능구조



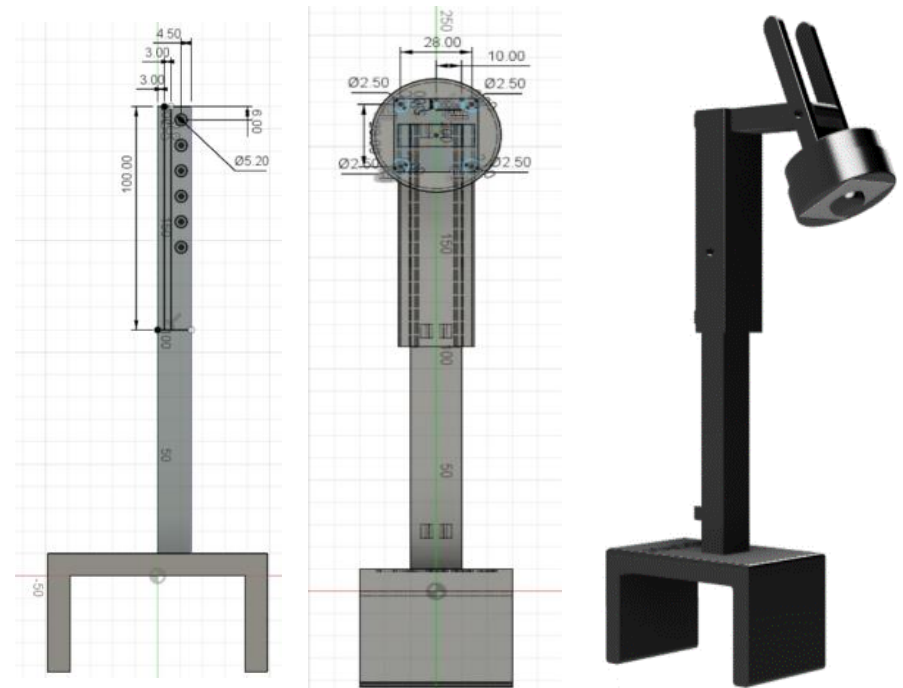
제품 구성도

- 구성 : 카메라, 부착 기구부, 산업용 PC
- 크기 : 50(W) x 200(L,가변)



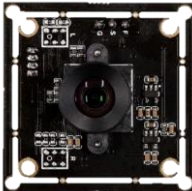


3D 설계도

※ 본체 3D 프린터용 도면 첨부



H/W 조립 준비물

제품단가 기준일 : 2023년 03월 01일

부품(시제품)	이미지	필요수량	단가	비고
HD 카메라		1개	89,500	모델명 : OV9782 브랜드 : Arducam 렌즈 : 광각 160도 서터유형 : 글로벌 서터
산업용 PC		1개	900,000	모델명 : HDL-T150PC-10CP CPU : i5 10210U Quad Core HDD : 120GB SSD 메모리 8GB DDR RAM 화면크기 : 15인치LED패널 터치방식 : 정전식 터치
본체(합체)		1set	20,000	3D프린터 출력 → 연구시 사용 모델 : Ultimaker S5 Pro 재료 : PLA 최소 충진률 : 40% # 주문생산 또는 자체생산 필수 - 설계도면 첨부 → [도면] 시편및제품자동판별합체

이미지 학습 방법

1. Yolov4를 이용하여 숫자 이미지 판독을 하기 위해 Google Colab를 이용하여 이미지를 학습시킨다.
Google Colab에서 Google Drive를 연결해준다.

```
[ ] from google.colab import drive
    drive.mount('/content/drive')
```

2. 작업을 진행할 폴더를 생성해주고 gitHub에서 darknet을 복사한다.

```
[ ] !mkdir yolov4
    %cd yolov4
    !git clone https://github.com/AlexeyAB/darknet
```

3. Darknet을 컴파일 하기 전 옵션을 설정해주고 컴파일을 진행한다.

```
[ ] %cd darknet/
    !sed -i 's/OPENCV=0/OPENCV=1/' Makefile
    !sed -i 's/GPU=0/GPU=1/' Makefile
    !sed -i 's/CUDNN=0/CUDNN=1/' Makefile
    !sed -i 's/CUDNN_HALF=0/CUDNN_HALF=1/' Makefile
    !sed -i 's/LIBS0=0/LIBS0=1/' Makefile
```

```
[ ] !make
```

이미지 학습 방법

4. 학습할 이미지를 오픈소스 라벨링 툴인 CVAT을 이용하여 라벨링을 진행하면 좌표값이 적힌 txt파일을 얻을 수 있다.

학습할 이미지와 txt 파일을 압축하여 Google Drive로 업로드하고 압축을 해제한다.



```
[ ] !unzip /content/drive/MyDrive/RESEARCH/binggrae/obj.zip -d /content/yolov4/darknet/data/obj
```

5. 데이터학습에 필요한 설정파일들을 작성하고 업로드해준다.

```
[ ] !cp /content/drive/MyDrive/RESEARCH/binggrae/obj.data /content/yolov4/darknet/data
!cp /content/drive/MyDrive/RESEARCH/binggrae/obj.names /content/yolov4/darknet/data
!cp /content/drive/MyDrive/RESEARCH/binggrae/yolov4-tiny-custom.cfg /content/yolov4/darknet/cfg
```

obj.data X

```
1 classes = 14
2 train = data/train.txt
3 valid = data/valid.txt
4 test = data/test.txt
5 names = data/obj.names
6 backup = backup/
```

obj.names X

```
1 classic_fe1.5
2 classic_nonfe2.5
3 classic_sus2.0
4 classic_sus2.5
5 fe_1.5mm
6 non-fe_2.5mm
7 plain_fe1.5
8 plain_nonfe2.5
9 plain_sus2.0
10 plain_sus2.5
```

yolov4-tiny-custom.cfg X

```
1 [[net]]
2 # Testing
3 #batch=1
4 #subdivisions=1
5 # Training
6 batch=64
7 subdivisions=16
8 width=416
9 height=416
10 channels=3
```

이미지 학습 방법

6. gitHub에서 YOLOv4 사전학습모델을 다운로드 한다.

```
[ ] #yolov4-tiny
!wget https://github.com/AlexeyAB/darknet/releases/download/darknet_yolo_v4_pre/yolov4-tiny.conv.29
```

7. 업로드한 data, cfg, weight 파일 경로를 지정하고 학습을 진행한다.

```
[ ] #yolov4-tiny
!./darknet detector train data/obj.data cfg/yolov4-tiny-custom.cfg yolov4-tiny.conv.29 -dont_show
```

8. 학습이 완료된 모델을 이용하여 학습시킨 이미지 또는 판독할 숫자 이미지를 업로드하고 간단한 python 코드를 실행하여 테스트를 진행해본다.

```
[ ] !./darknet detector test data/obj.data cfg/yolov4-tiny-custom.cfg yolov4-tiny-best.weights test.jpg
from IPython.display import Image
Image('result.jpg')
```

S/W 구동 방법

1. 첨부된 python-3.9.exe 파일을 설치한다.

※ python 3.9 버전으로 개발되어 다른 버전일 경우 수정이 필요할 수 있습니다.

2. Python 라이브러리를 별도로 모두 설치해주거나 첨부된 라이브러리 압축파일을 Python이 설치된 경로에 Lib 폴더 안에 복사해준다.

※ python 설치 경로 예시

C:\Users\사용자\AppData\Local\Programs\Python\Python39

3. 첨부된 시편및제품자동판별센서 소스코드(x_ai_vision.zip) 압축 해제 후 config 폴더에 이미지 학습을 통해 얻은 cfg파일과 weights 파일을 복사해준다.

4. Python 프로그램 툴을 이용하여 실행하거나 Windows cmd 프로그램에서 x_ai_vision_web.py 파일을 실행한다.

```
C:\Users >python C:\Users\x_ai_vision\x_ai_vision_web.py
camera-0 start

* Serving Flask app 'x_ai_vision_web'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000

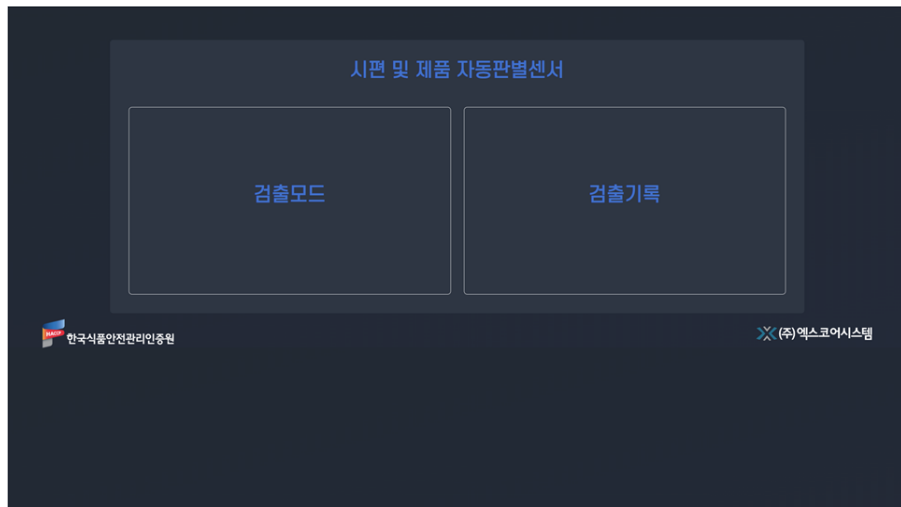
Press CTRL+C to quit
* Restarting with stat
```

5. 인터넷창에서 프로그램에 접속한다.

접속주소 : <http://127.0.0.1:5000> 또는 <http://localhost:5000>

구분	구현 기술	활용 영역
운영체제	• Windows 10	(권장 스택) OS 기반 운영체제 활용 권장
개발언어	• Python v3	(권장 스택) 가장 경량화된 개발 언어
비전	• YOLOv4-Tiny	오픈소스 기술 활용 (사전학습, 자동판별)
Library 활용	<ul style="list-style-type: none"> • Python pip 라이브러리 모듈 - flask - flask_socketio - opencv-python - numpy - flask_cors - pandas - threading - base64 - sqlite3 - requests - numba - scipy - filterpy 	운영체제별 오픈소스 Library 활용 가능

메뉴 선택 방법



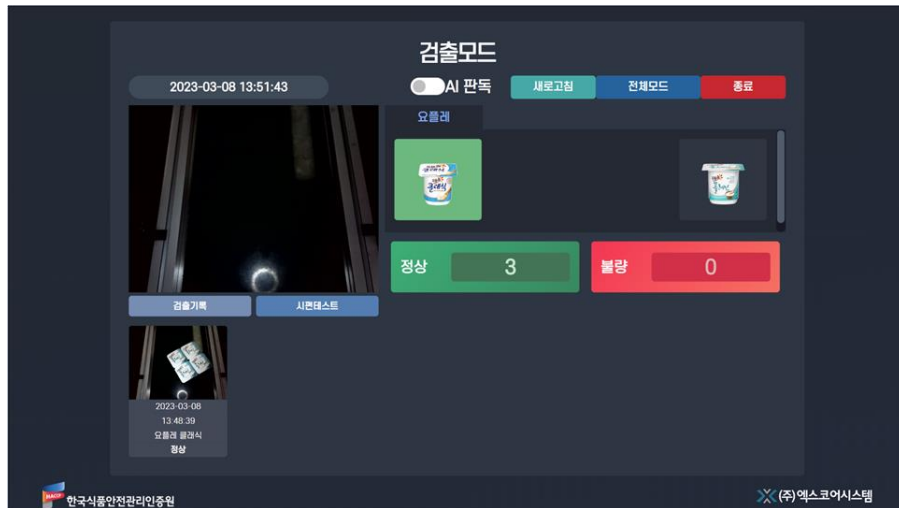
1. 프로그램이 실행되면 '검출모드', '검출기록' 2개의 메뉴가 보여집니다.
2. '검출모드' 메뉴를 선택하면 일반적인 금속검출 공정에서 사용하는 검출 모드와 주기적으로 시행하는 시편테스트 모드를 진행할 수 있습니다.
3. '검출기록' 메뉴를 선택하면 지금까지 금속검출 공정을 진행한 데이터 기록을 확인할 수 있습니다.

검출기록



1. 판별된 이미지와 제품명, 검출일시, 금속검출 여부, 시편테스트 여부, 이탈여부가 화면에 표시됩니다.
2. 하단 목록에 지금까지 진행된 금속검출 공정의 데이터가 나타납니다. 페이지 이동을 통해서 과거 데이터를 확인 할 수 있습니다. 목록 중 하나의 항목을 선택하면 상세정보가 상단에 나타납니다.
3. '검출모드' 버튼을 누르면 해당 메뉴로 이동할 수 있습니다. '종료' 버튼을 누르면 메뉴 선택을 할 수 있는 첫 화면으로 이동 합니다.

검출모드 화면

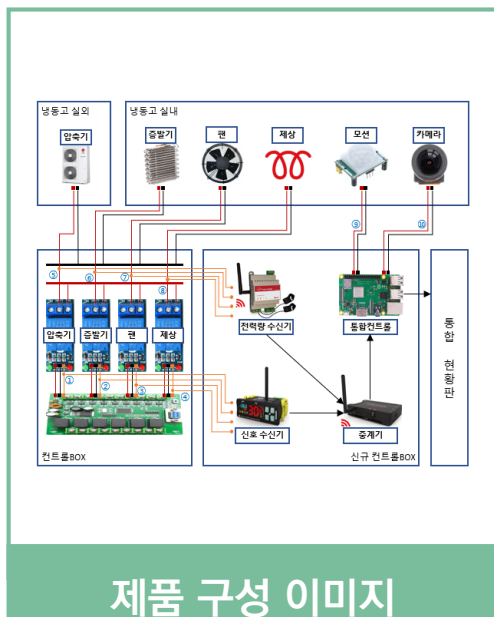


1. 실시간 영상이 왼쪽에 나타나고 오른쪽에 제품 목록이 표시됩니다.
현재까지 진행한 금속검출 공정의 정상/불량 개수가 표시되고 하단에 제품의 이미지와 검출시간, 제품명, 금속검출 유무가 리스트 형태로 나타납니다.
※ 제품 목록은 x_ai_vision 폴더 내에 x_ai_vision.db 파일을 통해서 수정할 수 있습니다.
2. “AI 판독” 기능을 끈 경우 선택된 제품으로만 카운팅 및 검출이 진행됩니다.
“AI 판독” 기능을 켜 경우 학습된 데이터 기반으로 제품을 자동으로 판독하여 카운팅 및 검출 유무를 판별합니다. 그리고 시편을 판독하는 경우 자동으로 “시편테스트 모드”로 변경됩니다.
3. “검출기록”, “시편테스트” 버튼을 누르면 해당 모드로 변경됩니다.

시편테스트 화면



1. 실시간 영상이 왼쪽에 나타나고 오른쪽에 시편테스트 진행해야할 목록이 표시됩니다.
※ 시편테스트 목록은 x_ai_vision 폴더 내에 x_ai_vision.db 파일을 통해서 수정할 수 있습니다.
2. “AI 판독” 기능을 끈 경우 진행한 제품 또는 시편을 직접 체크해야 합니다.
“AI 판독” 기능을 켜 경우 학습된 데이터 기반으로 제품 또는 시편 그리고 제품+시편을 자동으로 판독하여 체크됩니다.
등록된 시편테스트가 모두 진행된 경우 자동으로 “검출모드”로 변경됩니다.
3. “검출기록”, “검출모드” 버튼을 누르면 해당 모드로 변경됩니다.

기술의
특장점

- 냉장 및 냉동고의 온도변화와 한계기준이탈상황(제상, 작업중 등)의 자동식별 및 처리 가능 제공
- 냉장·냉동 설비의 정상 작동 여부 모니터링 기능 제공
- 기존 기술 및 제품 대비 구축, 운영 비용 70% 이상 절감

제품
구성 내역통합컨트롤Box
(제어부 포함)

온도센서

모션인식센서

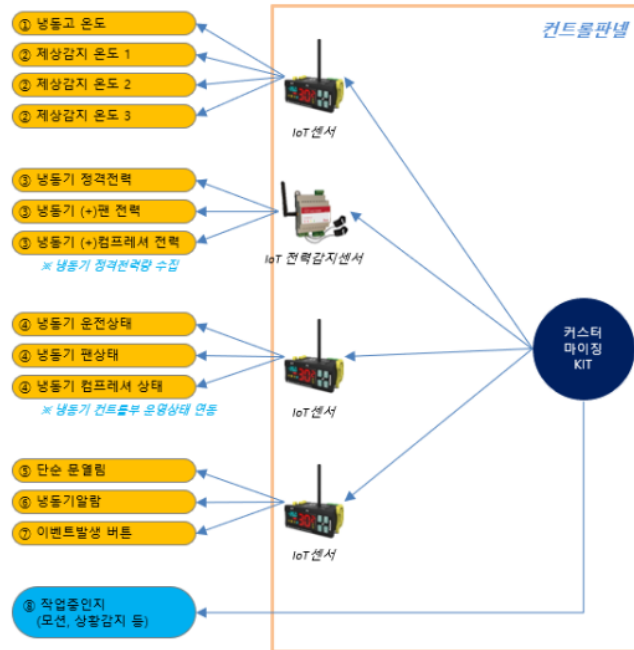
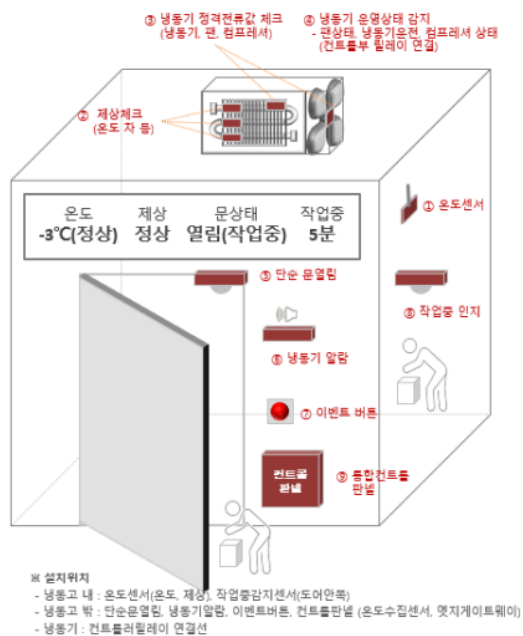
신호 수신기

전력량 수신기

데이터 중계기

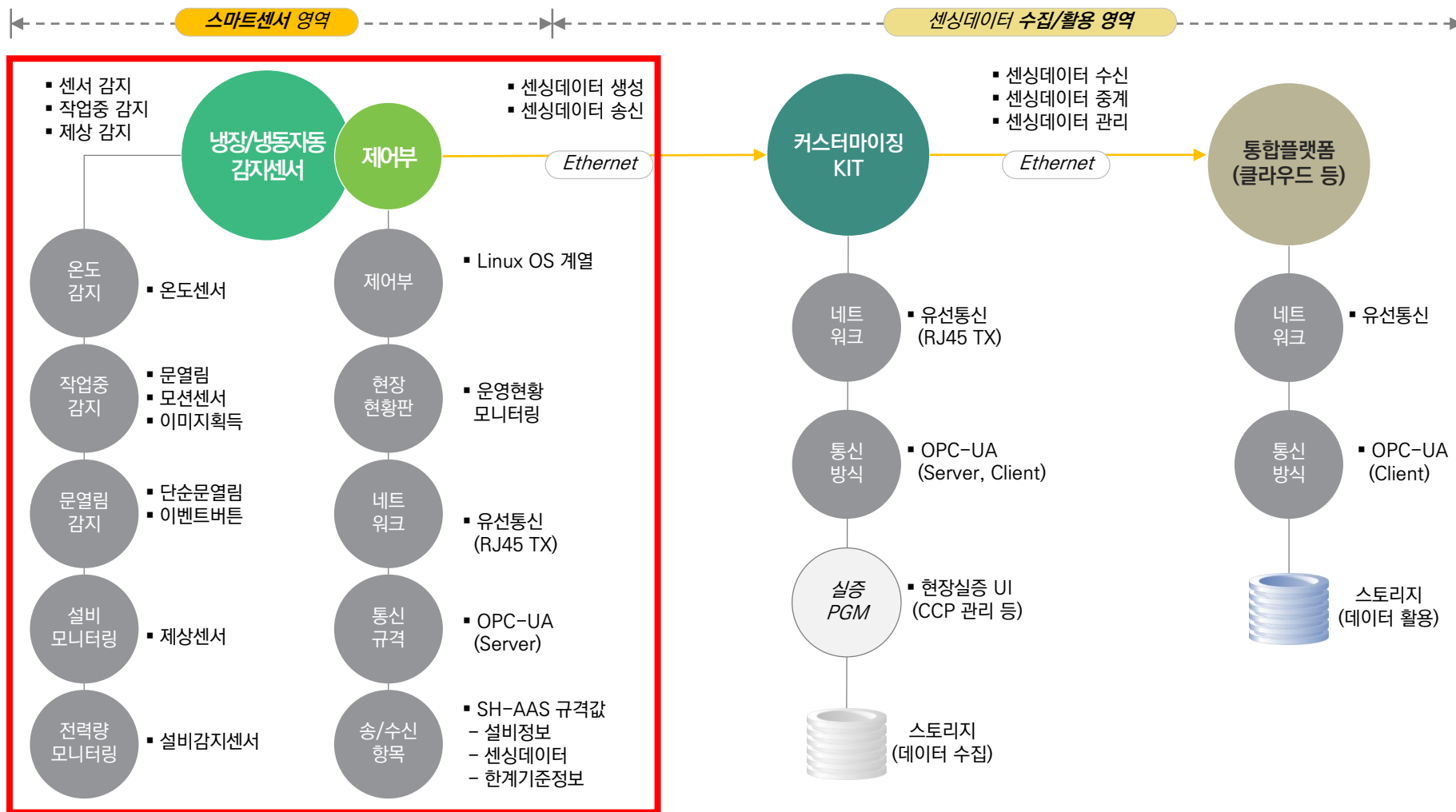
구현기술

구분	활용 기술	신규 개발	비고
구현 기능	<ul style="list-style-type: none"> 온도관리 설비모니터링 (컨트롤러 연동) 	<ul style="list-style-type: none"> 모션감지 통합관제 한계기준 자동예외처리 	작업장의 여러 관리요소에 센서 설치 후 통합하여 관리되는 구조로 연구개발
개발 요소	현 기능을 통합	신규 개발	

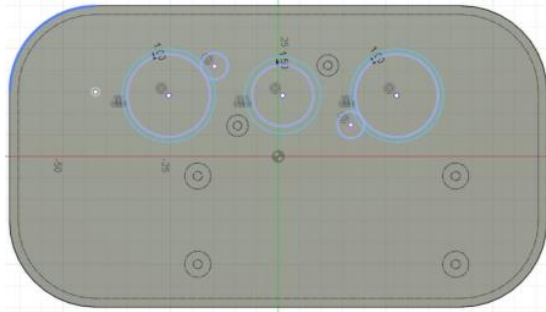


- 온도센서 : 냉장·냉동고 온도 감지
- 제상감지(온도)센서 : 제상 가동 상태 감지
- 전력감지센서 : 설비의 가동 여부 감지
- 컨트롤러신호감지센서 : 컨트롤러 릴레이
- 문열림감지센서 : 문열림 감지
- 알람 : 각종 이벤트 알람
- 이벤트버튼 : 이벤트 발생 버튼
- 모션센서 : 작업자 모션을 직접 감지
- 통합컨트롤박스 : 통합제어박스

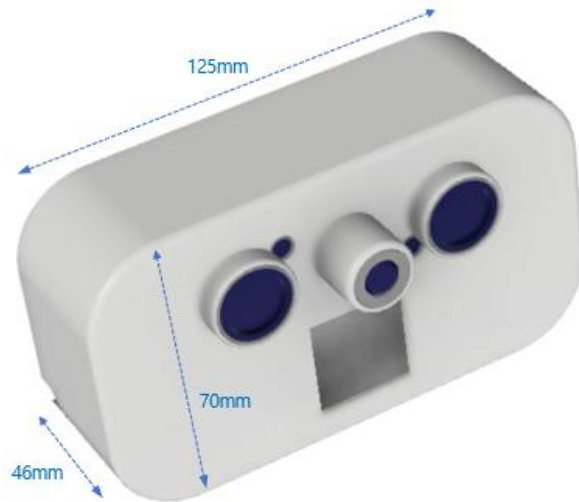
기능구조



모션감지센서 설계도




※ 본체 3D 프린터용 도면 첨부



H/W 조립 준비물

제품단가 기준일 : 2023년 03월 01일

부품(시제품)	이미지	필요수량	단가	비고
인체모션감지 센서 모듈 (PIR)		1개	25,000	모델명 : OSTSen-PIR100 브랜드 : 온시스텍 감지거리 : 최대 7m 감지 각도 : 110 ° (±55 °) 감지 주기 : 4초
카메라 모듈 (적외선 조광 NOIR)		1개	30,000	모델명 : YR-030 조리개값 : 1.8 촬영거리 : 야간 기준 5~8m 화각 : 60 °
산업용 PC		1개	900,000	모델명 : HDL-T150PC-10CP CPU : i5 10210U Quad Core HDD : 120GB SSD 메모리 8GB DDR RAM 화면크기 : 15인치LED패널 터치방식 : 정전식 터치





H/W 조립 준비물

제품단가 기준일 : 2023년 03월 01일

부품(시제품)	이미지	필요수량	단가	비고
라즈베리파이 4B 보드		1개	120,000	모델명 : 라즈베리파이 4B
본체(합체)		1set	15,000	3D프린터 출력 → 연구시 사용 모델 : Ultimaker S5 Pro 재료 : PLA 최소 충전률 : 40% # 주문생산 또는 자체생산 필수 - 설계도면 첨부 → [도면] 모션감지합체

H/W 조립 준비물

제품단가 기준일 : 2023년 03월 01일

부품(시제품)	이미지	필요수량	단가	비고
전력량 수신기		1개	550,000	제품명 : 무선 멀티 전력량계 측정채널 : 3 Channel or 12 Channel 전류센서 용량 : 20A ~ 400A 통신종류 : Zigbee/2.4GHz/modbus
방수형 온도센서		1개	10,000	제품명 : DS18B20 온도 범위 : -50℃~125℃ Voltage : 3.3V, 5V
신호 수신기		2개	350,000	제품명 : 신호 수신기 출력 : 릴레이 출력 4점 입력 : 온도센서 2점, 디지털 2점 통신종류 : Zigbee/2.4GHz/LAN
중계기		1개	400,000	제품명 : 중계기 출력 : 릴레이 출력 4점 입력 : 온도센서 2점, 디지털 2점 통신종류 : Zigbee/2.4GHz/LAN

구성도



주요 환경 설정

1. 고정 IP 설정

- dhcpd.conf 파일 수정
\$ sudo nano -i /etc/dhcpd.conf
- 고정 IP와 ROUTERS 정보 수정

```
# Example static IP configuration:
#interface wlan0
static ip_address=192.168.0.22
#static ip6_address=fd51:42f8:caae:d92e::ff/64
static routers=192.168.0.1
#static domain_name_servers=192.168.0.1 8.8.8.8 fd51:42f8:caae:d92e::1
```

2. 데이터 전송을 위한 환경 설정 파일 수정

- x_motion.py 파일 수정
\$ nano /home/haccp/x_motion.py
- 호출 url 및 parameter 값을 해당 서버 정보에 맞게 수정

```
image_binary = image_file.read()
encoded_string = base64.b64encode(image_binary)

motion_dict = {
    'motion_detect' : 1,
    'img_binary' : encoded_string.decode(),
    'time' : time.strftime('%Y-%m-%d %H:%M:%S')
}

motion_json = json.dumps(motion_dict)
motion_data = pickle.dumps(motion_json)
motion_data_len = pickle.dumps(len(motion_data))

requests.post("http://192.168.10.200:5000/motionDetect", data=motion_dict, timeout=3)
```

S/W 구동 및 접속 방법

1. 모션감지센서 전용 OS를 부팅이 가능하도록 준비

- 첨부된 OS 이미지를 micro-SD카드에 OS 형식으로 굽는다.
- OS 파일명 : smarthaccp_iot2_20230306.iso
 - ※ 모든 필수 설정과 필수 라이브러리가 설치되어 있는 OS 이미지 파일
- 첨부된 rufus-3.21p.exe 파일을 이용하여 준비한 micro-SD카드에 첨부된 OS를 설치한다.
 - ※ rufus-3.21p.exe 파일은 OS 이미지를 micro-SD로 쉽게 설치해주는 무료 프로그램

2. 무선 네트워크 설정

- 설치가 완료된 micro-SD카드에서 아래 파일을 무선 환경에 맞게 수정해준다.
- wpa_supplicant.conf

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
network={
    ssid="접속하고자 하는 wifi 이름"
    psk="접속하고자 하는 wifi 비밀번호"
    key_mgmt=WPA-PSK
}
```

3. 설치가 완료된 micro-SD카드를 라즈베리파이 제로 보드에 결합 후 전원버튼을 눌러 시스템을 구동한다.

4. 시스템 접속

- 무선 공유기에서 접속된 모션감지센서 IP를 확인한다.
 - ※ 보통 내부 네트워크 정보에서 단말기명이 "raspberrypi"를 찾으면 된다.
- ssh 접속 : port 22 / ID haccp / PW haccp

구분	구현 기술	활용 영역
운영체제	• Raspbian OS	Firmware 기반 변경 활용 가능
개발언어	• Python v3	가장 경량화된 HW 개발 언어
활용 Library	<ul style="list-style-type: none"> • 라즈베리파이 Library <ul style="list-style-type: none"> - serial - socket - pickle - base64 - picamera - Rpi.GPIO - gpiozero - sqlite3 	운영체제별 오픈소스 Library 활용 가능

S/W 구동 방법

1. Python v3과 python 코드를 작성할 프로그램을 설치한다.

※ python 3.9 버전으로 개발되어 다른 버전일 경우 수정이 필요할 수 있습니다.

2. Python 라이브러리를 별도로 모두 설치해주거나 첨부된 라이브러리 압축파일을 Python이 설치된 경로에 Lib 폴더 안에 복사해준다.

※ python 설치 경로 예시

C:\Users\사용자\AppData\Local\Programs\Python\Python39

3. Python 프로그램 툴을 이용하여 실행하거나 Windows cmd에서 프로그램을 실행한다.

```
C:\Users >python C:\Users\x_ai_ocr\x_ai_vision_web.py
camera-0 start
```

```
* Serving Flask app 'x_ai_vision_web'
```

```
* Debug mode: on
```

```
WARNING: This is a development server. Do not use it in a production
deployment. Use a production WSGI server instead.
```

```
* Running on all addresses (0.0.0.0)
```

```
* Running on http://127.0.0.1:5555
```

```
Press CTRL+C to quit
```

```
* Restarting with stat
```

4. 인터넷창에서 프로그램에 접속한다.

접속주소 : <http://127.0.0.1:5555> 또는 <http://localhost:5555>

구분	구현 기술	활용 영역
운영체제	• Windows 10	(권장 스펙) OS 기반 운영체제 활용 권장
개발언어	• Python v3	(권장 스펙) 가장 경량화된 개발 언어
비전	• YOLOv4-Tiny	오픈소스 기술 활용 (사전학습, 자동판별)
Library 활용	<ul style="list-style-type: none"> • Python pip 라이브러리 모듈 - flask - flask_socketio - opencv-python - numpy - flask_cors - pandas - threading - base64 - sqlite3 - requests - numba - scipy - filterpy 	운영체제별 오픈소스 Library 활용 가능

메인 화면



1. 메인 화면에서 여러 개의 냉동창고에 대한 정보를 확인할 수 있습니다. 각 냉동창고에 온도, 정상 온도 범위, 문열림 여부, 작업 감지 여부, 설비 상태를 확인할 수 있고 냉동창고를 선택하면 상세 화면으로 이동합니다.
2. 실시간으로 창고 정보가 갱신되고 이상 유무를 판단하여 사용자가 보다 쉽게 인식할 수 있도록 UI를 변경해줍니다.

※ 미작동 - 회색, 정상 - 초록색, 경고 - 노랑색, 이탈 - 빨강색

상세 화면



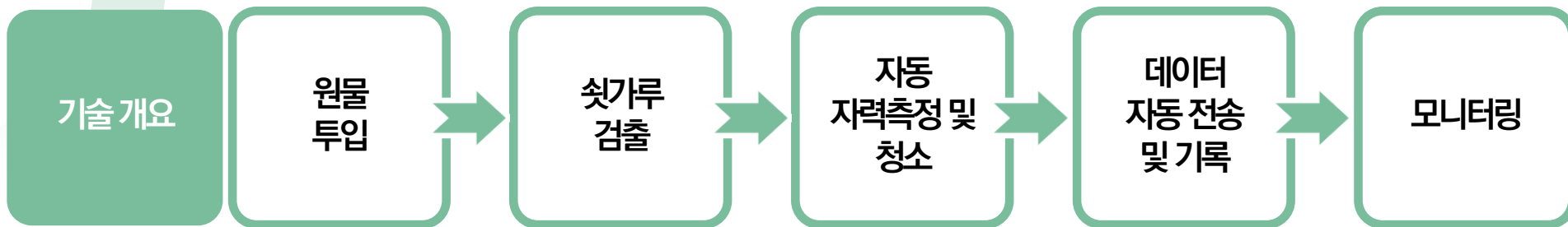
1. 현재 온도와 문열림, 작업여부 그리고 각 설비의 상태가 실시간으로 갱신됩니다. 각 데이터를 통해 자동으로 운영상태가 판별되고 모든 데이터가 갱신될 때마다 왼쪽 하단의 목록에 추가됩니다.

※ 운영상태 판별방법 참고

※ 하단 목록은 당일 데이터만 표시됩니다.

2. 각 센서 데이터 및 운영상태 변경 시 실시간으로 서버에 전송하여 활용할 수 있습니다.

※ refri_web 폴더 내 refri_web.py 파일에서 전송 서버 정보를 변경할 수 있습니다.



제품 이미지

기술의 특장점

- 수기로 이루어지던 자력 측정 업무를 자석봉 탈부착 없이 **자동으로 측정하여 데이터 기록**
- 췌가루 검출 후 **자동 청소 기능 제공**
- 기존 기술 및 제품 대비 구축, 운영 비용 **45% 이상 절감**

제품 구성 내역

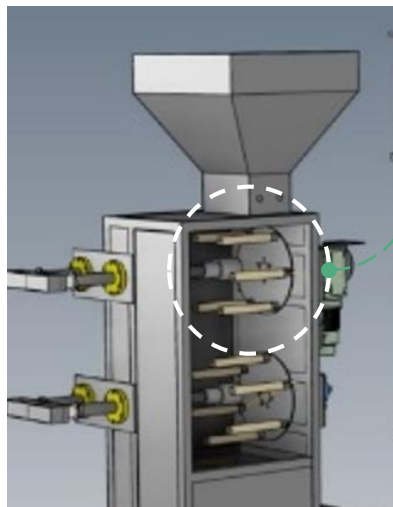
자석봉 자력 자동 측정을 위한
자력 측정부

췌가루 자동 제거를 위한
자동 청소부

제거된 췌가루량 자동 측정을 위한
무게 측정부

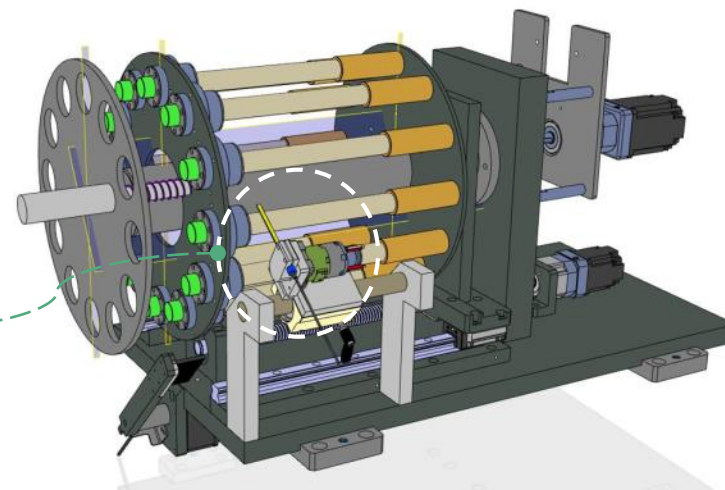
구현 기술

구분	현 기술	자동화 요소
자력측정	<ul style="list-style-type: none"> 작업자 수기측정 <ul style="list-style-type: none"> 네오돔방식 자석봉 사용 중 	<ul style="list-style-type: none"> 자력자동측정 <ul style="list-style-type: none"> 자동화 기구부 제작을 통한 자력자동측정 및 데이터 추출
검출된 섯가루 청소	<ul style="list-style-type: none"> 작업자 수기작업 <ul style="list-style-type: none"> ※ 일부 자동청소 사용 중 (데이터 자동기록 X) 	<ul style="list-style-type: none"> 자동 청소 및 자동 기록 검출섯가루량 자동 측정

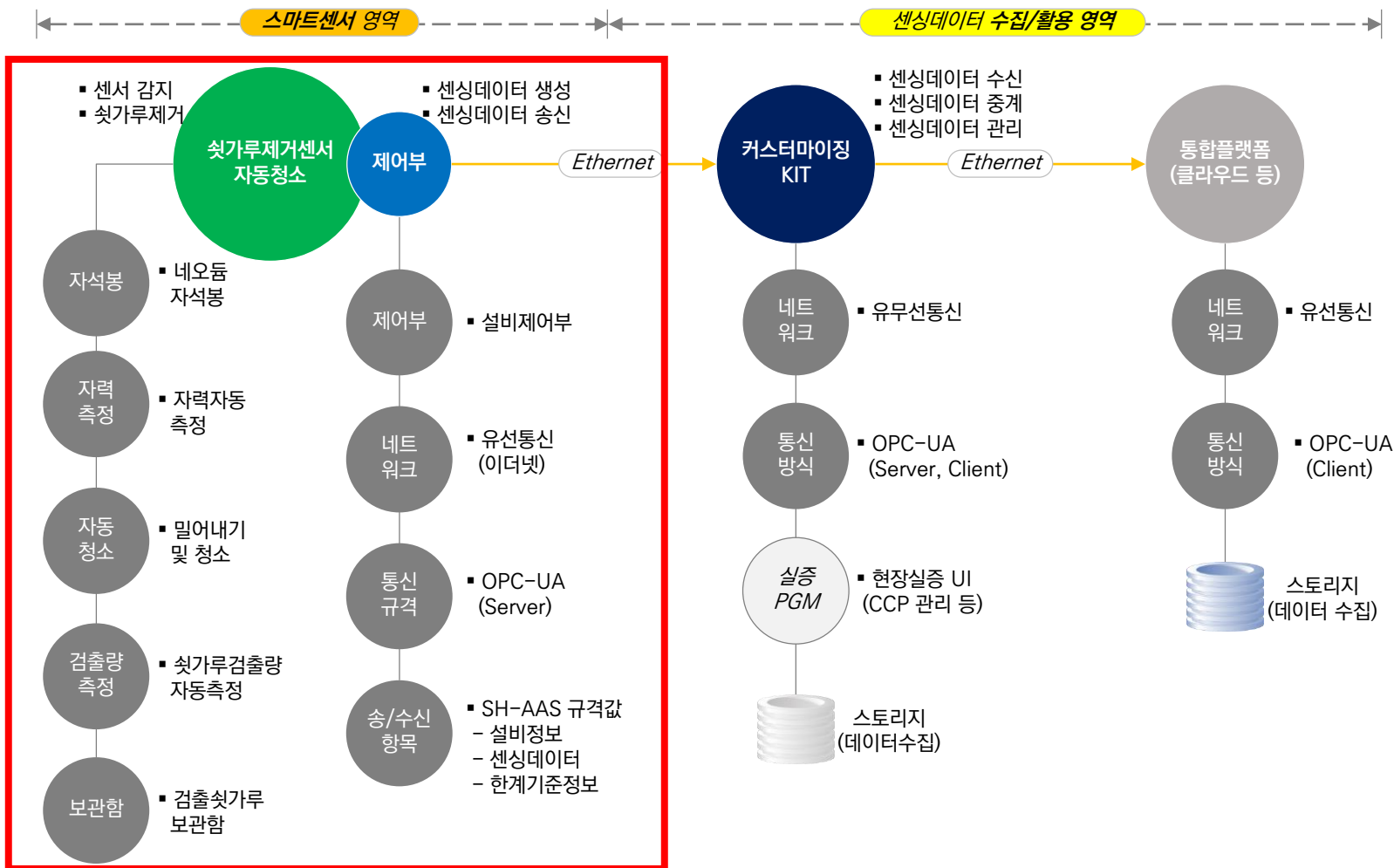


✓ 가장 대중적인 로터리자석봉 방식 활용

✓ 탐침프로브 기구자동화를 통한 자력자동측정 기술 개발

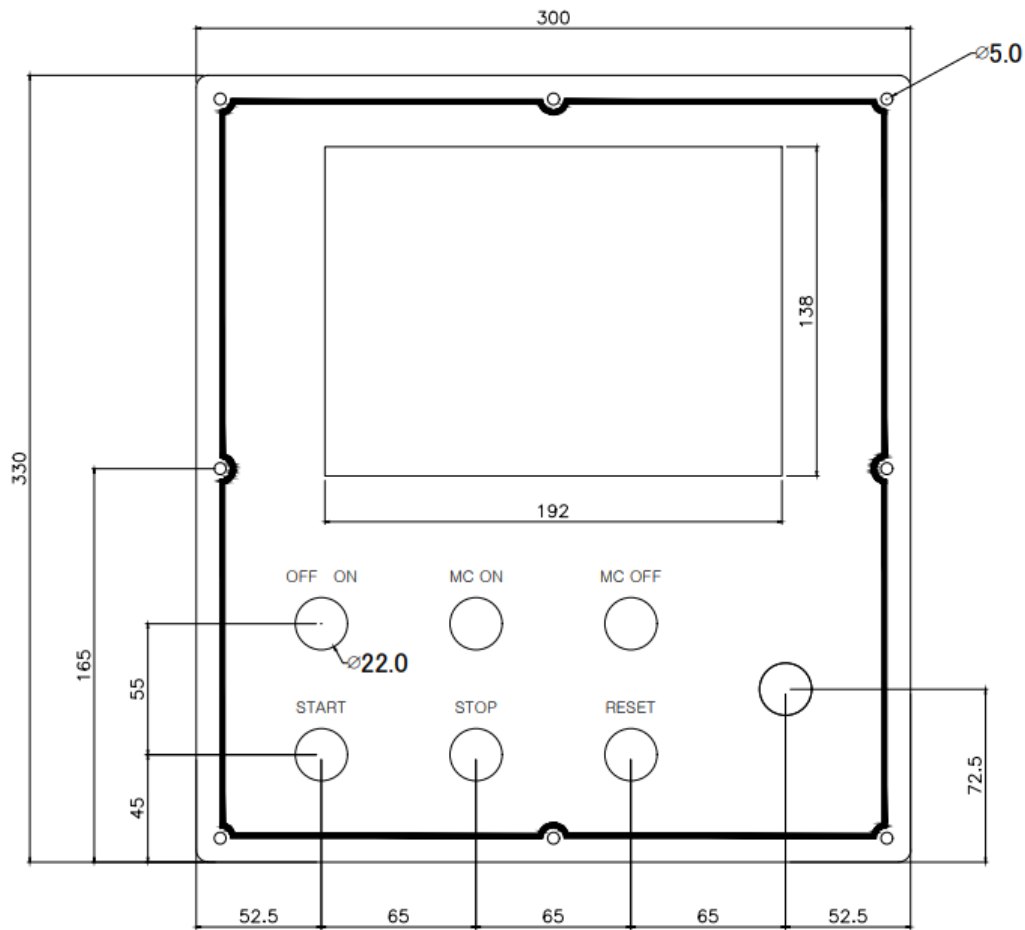


기능구조

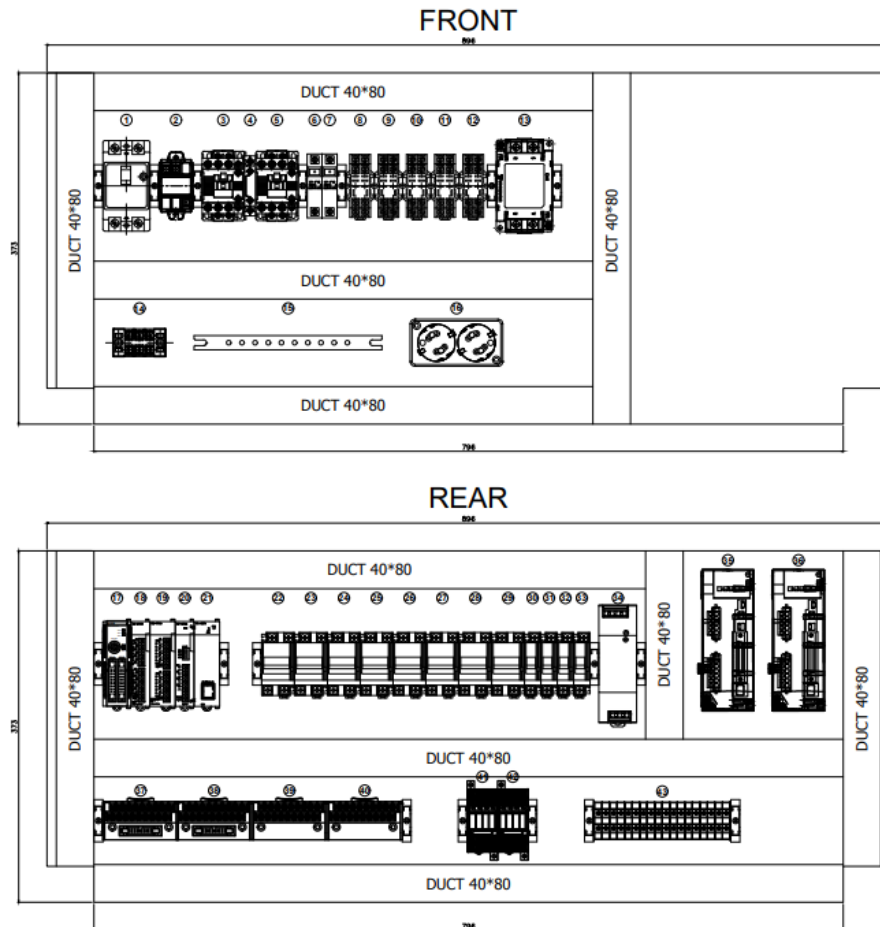


OP명판 도면

※ 전체 도면은 첨부



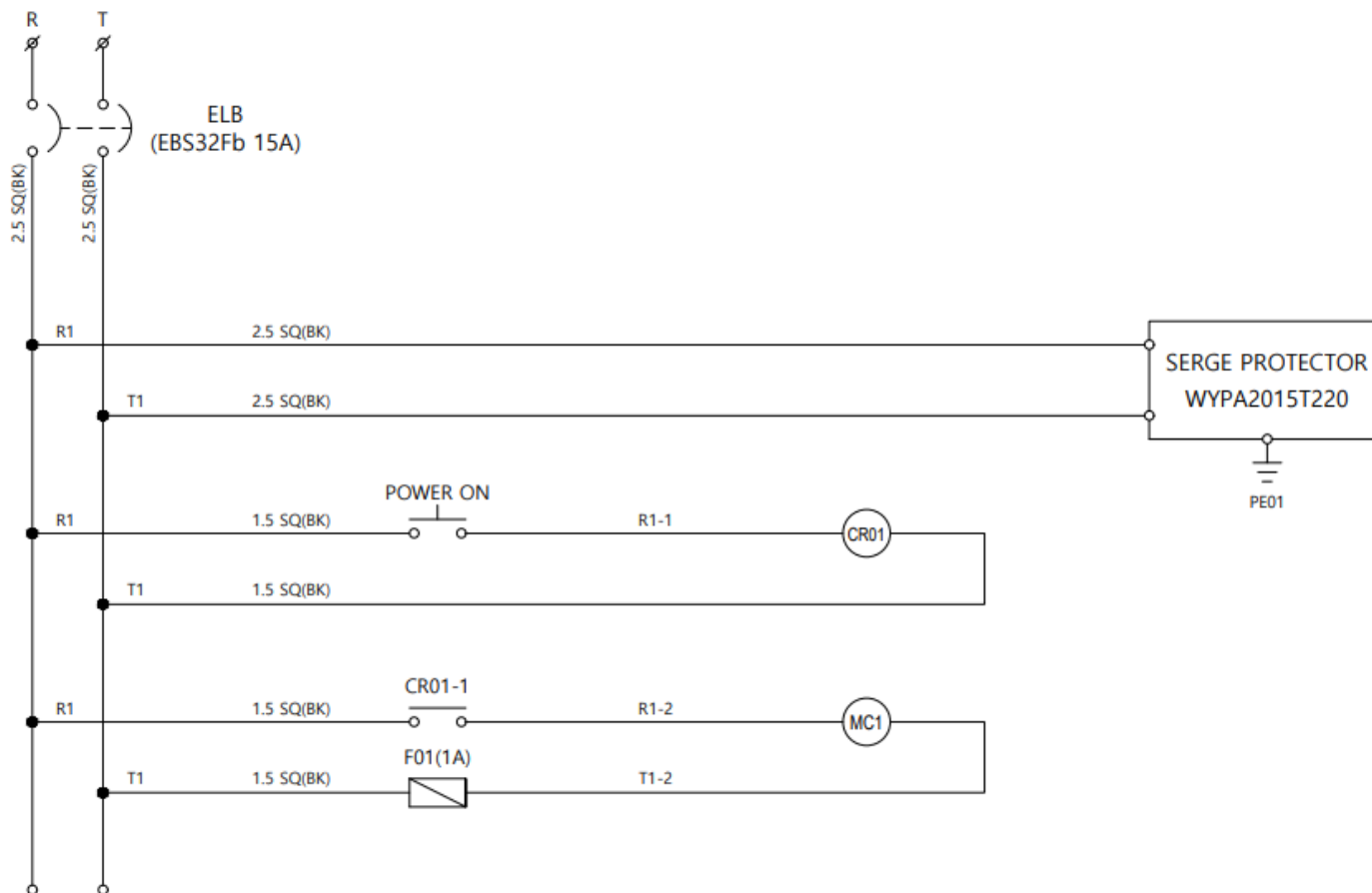
전장 배치도



번호	품명	규격	용도
①	ELB	EB32Fb 15A	MAIN POWER
②	SPD	WYP201ST220	SURGE PROTECTOR
③	MC1	MC-18b AC220V	MACHINE POWER LINE
④	보조퓨즈	UA-1	MC2 보조퓨즈
⑤	MC2	MC-18b AC220V	SERVO POWER LINE
⑥	F01	HY-F15-1A	MC1 COIL FUSE
⑦	F02	HY-F15-1A	MC2 COIL FUSE
⑧	CR01	S2R-LY2-N1 AC220	MC1 ON/OFF
⑨	CR02	S2R-LY2-N1 DC24	MC2 ON/OFF
⑩	CR03	S2R-LY2-N1 DC24	AC MOTOR #1 ON/OFF
⑪	CR04	S2R-LY2-N1 DC24	AC MOTOR #1 ON/OFF
⑫	CR05	S2R-LY2-N1 DC24	PC ON/OFF
⑬	N/F	WYF31ST1AD	NOISE FILTER
⑭	MAIN T/B	KT81-02003	MAIN TERMINAL BLOCK
⑮	EARTH BAR		EARTH
⑯	OUTLET		OUTLET
⑰	PLC	XBM-DK32S	PROGRAM
⑱	INPUT	XBE-DC16A	INPUT CARD
⑲	OUTPUT	XBE-FY16A	OUTPUT CARD
㉑	A/D CARD	XBF-AD04A	ANALOG CARD
㉒	이/디/엔 CARD	XBL-EMTA	이/디/엔 통신
㉓	CP01	LCP-32FM-3A	SMP5 POWER
㉔	CP02	LCP-32FM-1A	SERVO CONTROL PWR
㉕	CP03	LCP-32FM-5A	PC & MONITOR PWR
㉖	CP04	LCP-32FM-5A	SERVO R-AXIS PWR
㉗	CP05	LCP-32FM-5A	SERVO Z-AXIS PWR
㉘	CP06	LCP-32FM-2A	AC MOTOR #1 PWR
㉙	CP07	LCP-32FM-2A	AC MOTOR #2 PWR
㉚	CP08	LCP-32FM-7A	총 5.71 PWR
㉛	CP09	LCP-31FM-1A	TOUCH POWER
㉜	CP10	LCP-31FM-2A	I/O POWER
㉝	CP11	LCP-31FM-1A	PLC POWER
㉞	CP12	LCP-31FM-1A	MOTION POWER
㉟	SMP5	NDR-120-24	DC POWER
㊱	R-AXIS	MBDLT25SF	SERVO DRIVER
㊲	X-AXIS	MBDLT25SF	SERVO DRIVER
㊳	INPUT I/F	XTB-20H	INPUT INTERFACE
㊴	OUTPUT I/F	XTB-20H	OUTPUT INTERFACE
㊵	XTB-COM1	XTB-COM20B	P24.2, N24 COMMON
㊶	XTB-COM2	XTB-COM20B	P24.4, N24 COMMON
㊷	T.RY01	R4T-G8DH	CYLINDER SOL
㊸	T.RY02	R4T-G8DH	TOWER LAMP
㊹	T/B	D5TB-15A	TERMINAL BLOCK

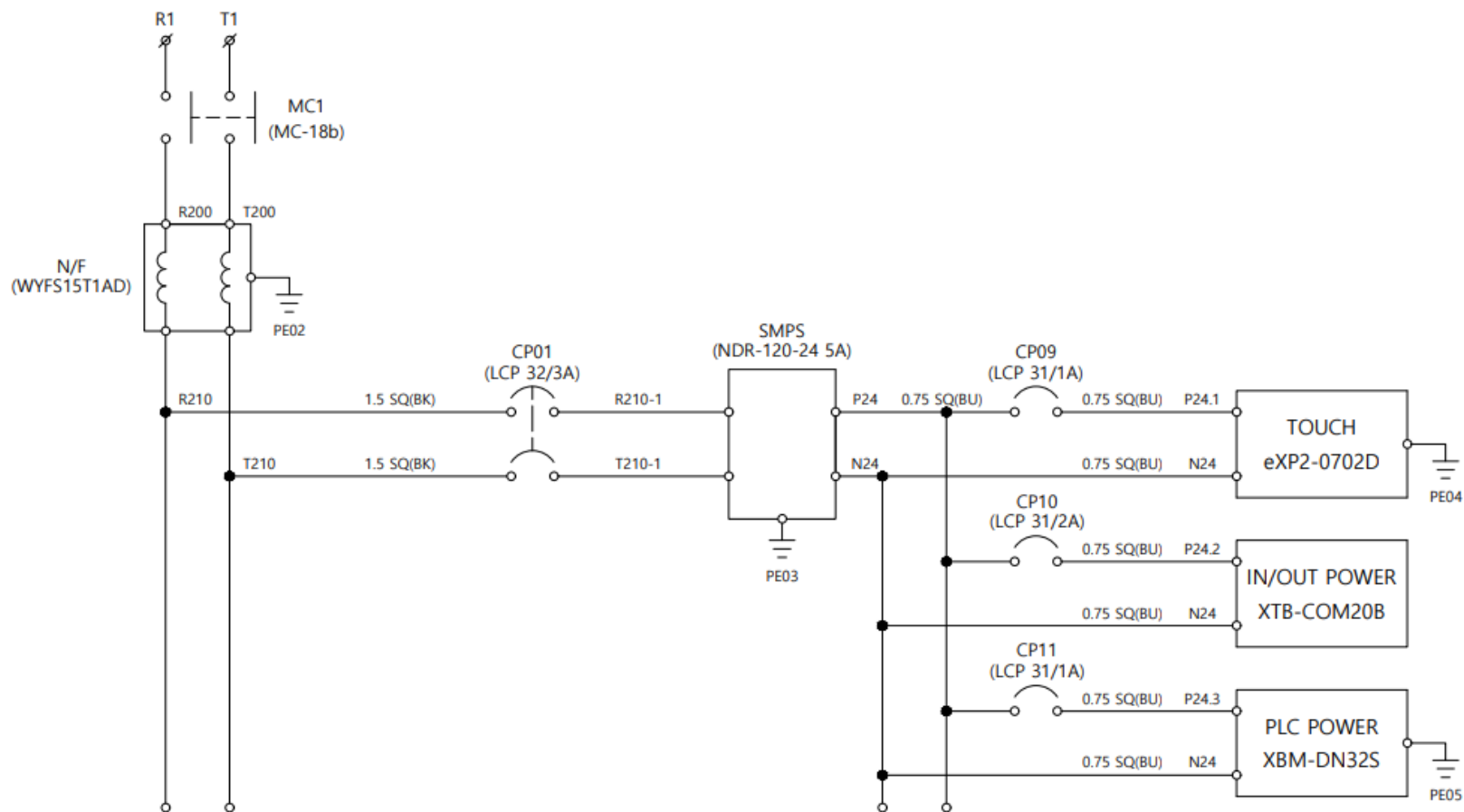
전장도면 1

MAIN 단상 220V

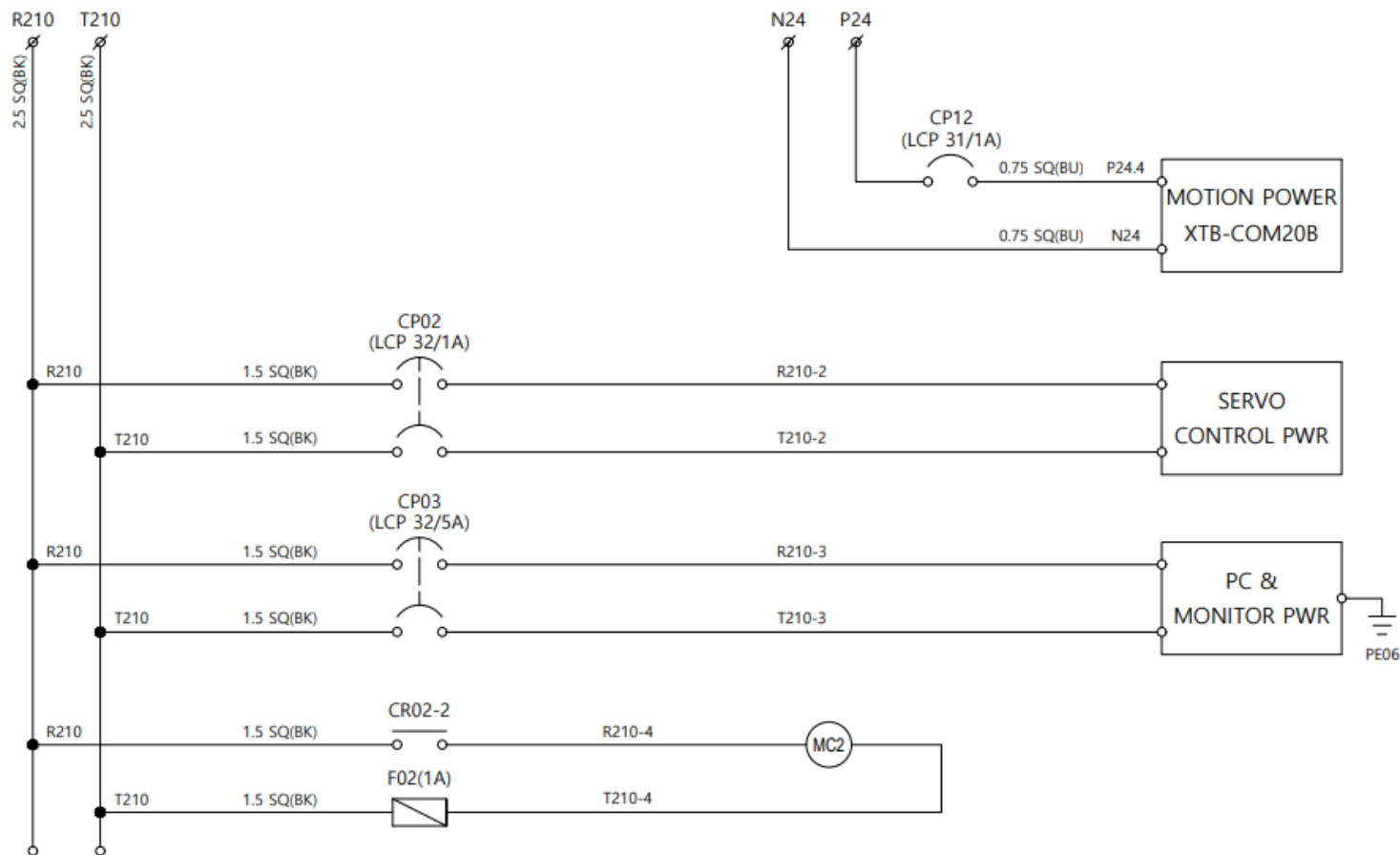


씻가루공정자동화센서 ▶ 설계도면

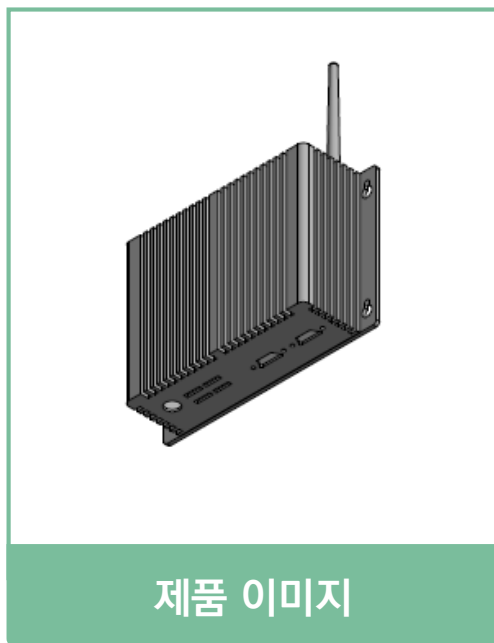
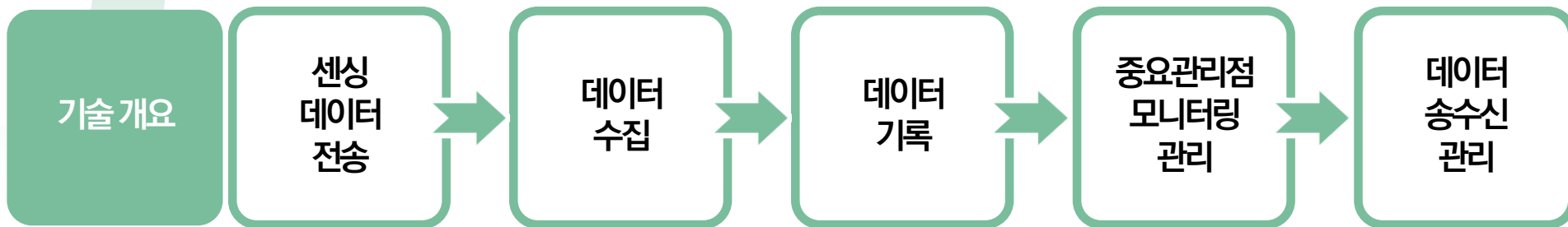
전장도면 2



전장도면 3



06 커스터마이징 KIT ▶ 기술 개요(1/2)



기술의 특장점

- 표준화가 적용된 스마트HACCP 기반 **스마트센서 데이터를 하나의 장치에서 처리/관리 및 모니터링**
- 설비 및 센서의 통신 중계, **스마트HACCP 표준모듈시스템 자동 연동**

제품 구성 내역

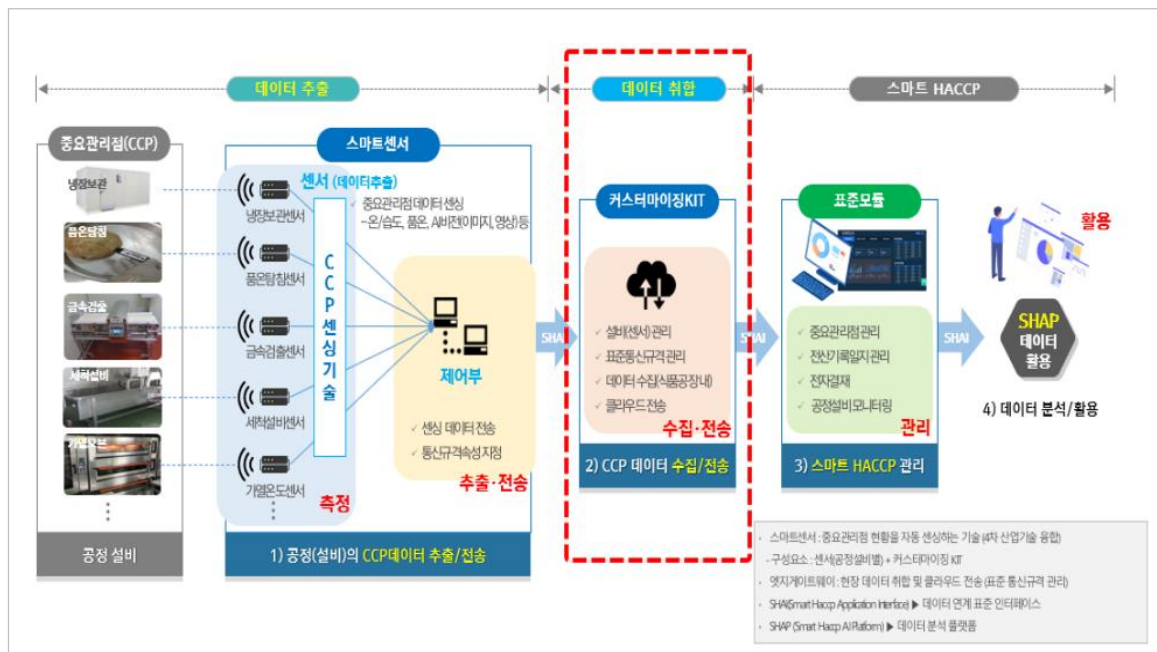
센서에서 수집되는 데이터 처리 및
관리를 위한
산업용 미니PC

중앙 제어와 데이터 처리
및 모니터링 위한
데이터처리모듈

06 커스터마이징 KIT ▶ 기술 개요(2/2)

구현 기술

구분	연구개발 요소	비고
데이터 인터페이스	<ul style="list-style-type: none"> 현장의 다양한 스마트센서 데이터 연동 유·무선 데이터 네트워킹 유연한 확장성 	네트워킹 우선순위 1순위) 무선 (Wifi, 블루투스 등) 2순위) 유선 (시리얼, 이더넷 등)
설치 용이성	<ul style="list-style-type: none"> 현장 작업장에 손쉬운 설치가 가능 현장 작업장에서 운영이 가능한 내구성 	작업장에서 안정적 운영을 위한 합체 제작



- ✓ 데이터 수집을 위한 표준통신규격 속성값 관리
- ✓ 현장의 설비 데이터를 자동 추출하는 스마트센서 데이터 수집
- ✓ 클라우드로 데이터 자동 전송

06 커스터마이징 KIT ▶ 기능구조

