Github

- 소스코드 버전 관리는 github에서 하고 있음
- 개발용 계정이 github@tricubics.com로 사용하고 있음
- 서비스 배포용 gitlab@tricubics.com 로 사용하고 있음
- eztc1234@@
- 두 계정의 비밀번호는 eztceztc1234 임
- github에 있는 프로젝트
    - lightserver: 센서를 제외하여 다른 장비 혹은 외부하고 통신할 때 사용함
    - system: 센서를 제어하는 프로그램
    - shell: 주기적으로 이미지와 로그를 삭제하는 shell script
      재부팅할 때 자동으로 system와 lightserver를 실행하는 파일
    - sensorTest: 냉장고를 설치할 때 센서를 테스트하는 프로그램
        - CamIndexTest: 카메라 index를 확인하는 프로그램
        - CamParamsTest: 카메라 파라미터를 확인하는 프로그램
        - CameraMonitor: 카메라 테스트 프로그램
        - LoadCellMonitor: 로드셀 테스트 프로그램
        - LoadCellNumberSetup: 로드셀 번호를 설정 프로그램
        - DoorLockTester: 락장치 테스트 프로그램
        - newIoBoard: 신형 IO보드에 락장치와 로드셀 같이 테스트 하는 프로그램
    - lightserver_freezer: 냉동용 lightserver 프로그램임
    - system_freezer: 냉동용 system 프로그램임
- 효성 키오스크 프로그램이 파일 사이즈가 40MB 초과해서 Gitub에 올릴 수 없음
    - 소스 코드 위치는 hp laptop에 /home/trc/AndroidStudioProjects/SmallMonitor 에 있음

https://finds.synology.com/

NAS Server

- https://finds.synology.com/
- NAS Server
-

냉장고

시스템:

- 개발 언어와 프레임워크: C++, QT로 만들었음
- 개발 툴: QT creator
- 카메라, 무게센서, 락장치 등 장비를 제어하는 프로그램임
- 소스 코드 설명서:
    - *SmartVM.cpp:*
      프로그램의 main controller로 볼 수 있음
      각 모듈을 관리하는 역할

    - *WeightManager.cpp:*
      모든 로드셀을 제어하는 역할
    - WeightManager.cpp:
      실제로 IO 보드하고 데이터를 주고 받는 역할

    - *CameraManager.cpp:*
      모든 카메라를 제어하는 역할
    - *V4LDevice.cpp:*
      카메라 설정에 관련된 부분
    - *V4LGrabber.cpp:*
      카메라마다 thread에 이미지를 캡쳐하는 데 관련된 부분
    - *ImageWriter.cpp:*
      메모리에 있는 이미지 데이터를 이미지 파일로 저장하는 부분
    - *ParamsMonitor.cpp:*
      카메라 파라미터를 모니터링을 하고 초기화되면 재 설정하는 부분

    - *HttpsManager.cpp:*
      L/S하고 통신하는 부분, ZMQ를 사용하고 있음

    - *AudioManager.cpp:*
      speaker를 통해 mp3파일을 재생하는 부분

    - *Instance.cpp, common/Config.cpp:*

설정 파일(settings.ini)를 읽거나 쓰기하는 부분

- *common/QtLogWrapper.cpp:*
  로그를 스크린에 display하거나 파일로 저장하는 부분

- *build/play_audio.sh:*
  mp3파일을 재생하는 파일

- *build/find_serial.sh:*
  카메라 혹은 io board usb 포트가 끊어지면 장치 관리자에 해당 device를 찾는 shell script이다

- *build/settings.ini:*
  카메라, 로드셀, io board에 대한 설정파일

LightServer(L/S):

- 개발 언어와 프레임워크: javascript, nodejs로 만들었음
- 개발 툴: vscode
- 서버 통신 혹은 단말기 연동하는 프로그램
- 소스 코드 설명서:
  - *package.json:*
    필요하는 라이브러리를 관리하고 프로그램의 입구를 지정해 줌

  - *main.js:*
    프로그램의 main controller로 볼 수 있음
    각 모듈을 관리하는 역할

  - *webClient.js:*
    산덴 웹페이지 서버에 통신하는 역할

  - *visionClient.js:*
    AI서버하고 통신하여 무게정보를 보내기
    AI서버하고 통신하여 인식 결과를 수신하기
    사이드 이미지를 AI서버로 업로드 하기

- *testZMQ.js:*
  icebox.js를 테스트하기 위한 코드

- *play_audio.sh:*
  mp3파일을 재생하는 shell script파일

- *lotteTest.js:*
  lotteClient.js를 테스트하기 위한 코드이고 한쪽 냉장고만 테스트할 수 있음

- *lotteTest2.js:*
  lotteClient.js를 테스트하기 위한 코드이고 양쪽 냉장고 다 테스트할 수 있음

- *lotteClient.js:*
  workshigh키오스크하고 통신하는 부분이고 TCP/IP socket사용하고 있음
  TCP/IP socket client 버전
  TCP/IP socket server 버전

- *logger.js:*
  날짜별로 로그를 파일로 저장하거나 스크린에 display 하는 부분

- *kioskServer.js:*
  효성을 위해서 태블릿 pc와 통신하는 부분
  사용자가 냉장고를 점유하고 있는 걸 송신
  락 상태를 송신
  구매 리스트를 송신

- *kiccServer.js:*
  kicc단말기와 통신하는 부분
  token 발행 요청, 승인 요청, 결제 취소 요청
  일반 모드와 성인 인증 모드가 있음

- *kiccClient.js:*
  kiccServer.js를 테스트 위한 코드

- *imageSender.js:*
  top이미지를 AI서버로 업로드하는 부분

top이미지 업로드하기 끝나면 무게 정보를 송신

- *icebox.js:*
  시스템하고 통신하는 부분
  문을 열어 달라고 요청을 송신
  문이 닫으면 해당 이벤트를 수신

- *deviceInfo.json:*
  냉장고에 관련된 설정 파일
  AI 서버의 ip 주소
  선반 개수
  냉장고ID
  결제 취소 여부
  고객사
  고객사가 롯데인 경우 debug모드 disable여부 및
  냉장고 위치 하고 소켓 통신중 담당하는 역할

- *dbHelper.js:*
  C/S한테 해당 냉장고의 상품 리스트를 요청하는 부분

- *csClient.js:*
  C/S하고 통신하여 구매정보 송신
  이미지를 zip로 압축해서 업로드 하기
  C/S에서 문을 여는 요청을 수신
  불완전 거래에 대한 결제 요청을 수신
  결제 취소 및 재 결제 요청을 수신

- *cancelPayment_new.js:*
  결제 안되는 경우 수동으로 결제하는 부분

- *config.js*
  단말기, C/S서버, AI서버, 웹서버에 대한 정보들

- *audioPlayer.js*
  javascript에 shell script를 실행해서 mp3파일을 재생하기
  재부팅 하기

설치 방법**:**

- 보내 드린 setup.sh 파일을 이용하여 설치할 수 있음
    - sudo bash setup.sh

VisionServer(V/S)

- 개발 언어와 프레임워크: javascript, nodejs로 만들었음
- 개발 툴: vscode
- L/S와 통신하여 이미지를 수신해서 저장하기
  해당 이미지 폴더 정보와 무게정보를 함께 L1한테 전달
  인식 결과를 L/S한테 송신
- 소스 코드 설명서:
    - *package.json:*
      필요하는 라이브러리를 관리하고 프로그램의 입구를 지정해 줌

    - *server.js:*
      프로그램의 main controller로 볼 수 있음
      각 모듈을 관리하는 역할
      사이드 이미지를 수신하고 저장하는 부분
      L/S를 통신하기

    - *zmqManager.js:*
      ZMQ를 이용하여 L1,L3와 송, 수신함
      이미지 경로 또는 무게 정보를 L1한테 송신
      인식 결과를 수신

    - *logger.js:*
      날짜별로 로그를 파일로 저장하거나 스크린에 display 하는 부분

    - *imgReceiverStream.js:*
      top 이미지를 수신하기

    - *dbHelper.js:*
      C/S한테 냉장고 상품 리스트를 요청하는 부분

    - *config.js:*

L1, L3, L/S, C/S에 대한 ip 혹은 port 정보

- *deviceInfo.json:*
  냉장고마다 어떤 weight file를 사용하는지 지정하기

냉동고

냉동고는 냉장고와 유사하며 top카메 한개 더 추가함
사이드 이미지를 업로드 부분을 삭제됨

Open Store

LightServer(L/S):

- 개발 언어와 프레임워크: javascript, nodejs로 만들었음
- 개발 툴: vscode
- KICC단말기와 연동
  trc C/S와 연동
  C/S와 연동
  AI Center와 연동
- 소스 코드 설명서:
  - 냉장고 L/S하고 유사함
  - *storeClient.js:*
    AI Center와 연동하여 사용자 입장 시간정보 송신 및 퇴장 시간정보 수신
    AI Center와 연동하여 사용자 퇴장 시간정보 및 구매정보 수신
  - *storeServer.js:*
    storeClient.js를 테스트하는 코드
  - *aiClient.js:*
    trc C/S와 연동하여 불완전 거래 발생시 시간정보를 송신

설치 방법:

- 보내 드린 setup.sh 파일을 이용하여 설치할 수 있음
  - sudo bash setup_freezer.sh