

TOF LIDAR Sample KT-1

2021. 3. 16.
HYBO – 손영빈

1. Sample #KT-1.x

구현 사항

- Depth image generation
 - 320x80 (FOV H/V ~ 120° x 30°, resolution H/V ~ 0.375°)
 - 차후 아래 사항 구현을 위해 vertical resolution 은 0.75° 정도로 조정될 수 있음
- Grayscale image capture
 - 320x160 (calibration pattern capture 를 위한)
- Serial comm. I/F (12Mbaud)
- Optical power output ~ 80%
- TOF shutter speed 200usec (15klx ambient light 까지 대응 가능할 것으로 예상)

미 구현 사항

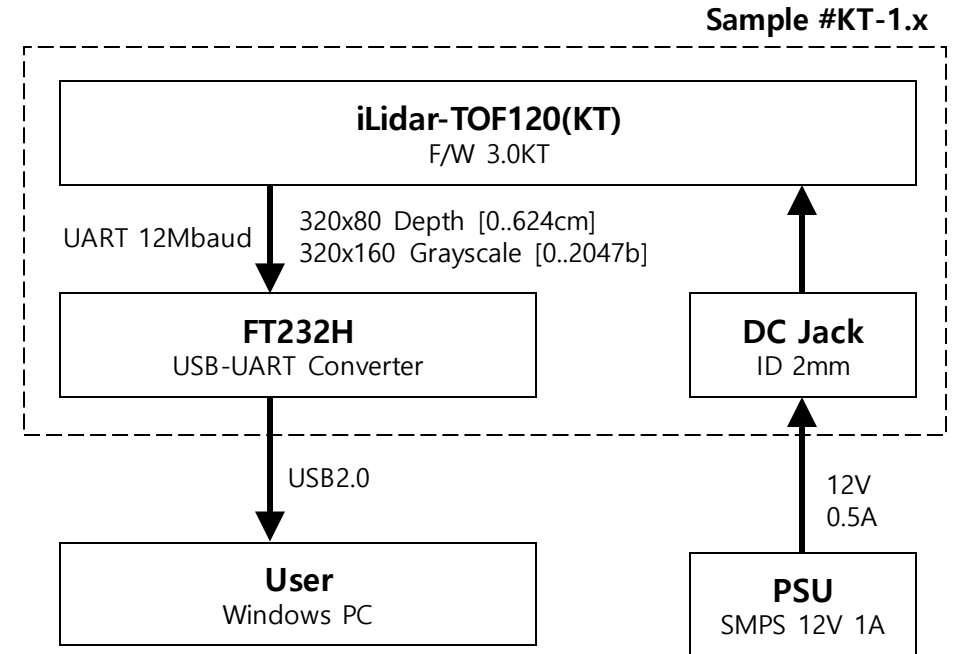
- Tx lens → 중앙부 5~6m, 좌우 가장자리 1m 까지 측정됨
- Integer ambiguity resolution → 0m, 6.25m, 12.5m, ... 형태로 구별 불가
- Sensor calibration → FPN/ambient light 대응 불가능하여 노이즈 심함
- Camera calibration → 3D 좌표계로 표현 안됨
- High dynamic range → 가까운 거리 측정 불가
- Motion blur reduction → 센서 or 피사체 움직일 시 blur 발생 심함
- Ethernet I/F → 현재 serial 통신만 사용
- Enclosure → 현재 Hybo 타입 enclosure를 개조하여 사용
- Synchronization → 센서 동시동작 불가

기타 사항

- Window 접착제 도포 불량 → 동작에는 큰 지장 없음



Sample #KT-1.x



Block Diagram

1. Sample #KT-1.x

미 구현 사항 – Tx lens

- Tx lens가 부착되지 않아 좌우측으로 나가는 신호가 엔클로저에 막힘
- 좌우 가장자리 방면 측정거리가 짧고 노이즈가 심함
- 현재 좌우 $\pm 50^\circ$ 이상은 1m 수준까지만 측정되는 것으로 보임

→ 4월 초~중순 경 개발 완료 예상

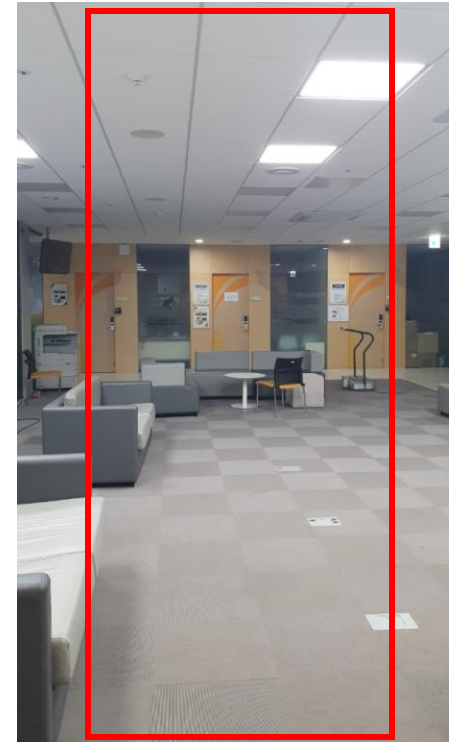
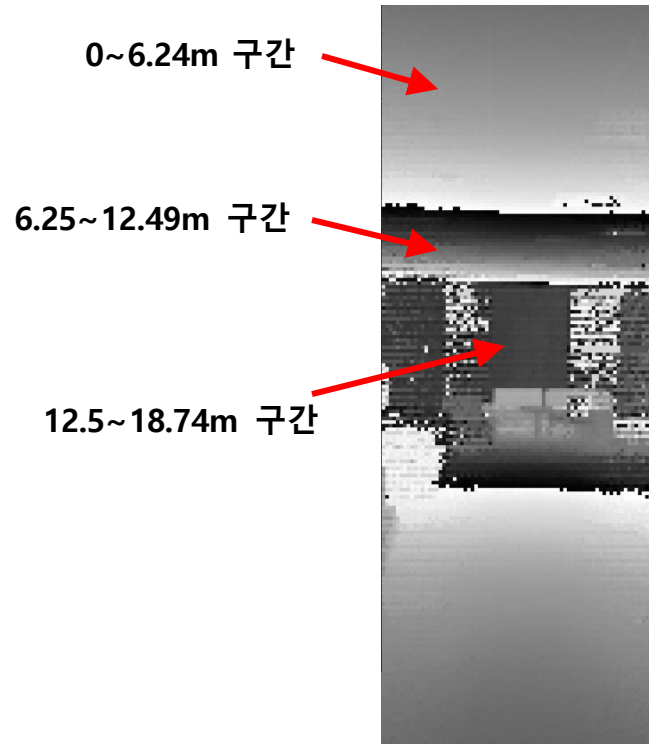
미 구현 사항 – Integer Ambiguity Resolution

- Rollover 된 거리끼리 구별 불가
- Ex) 0m 부터 점점 멀어지는 경우, 6.24m을 지나가면 다시 0m 부터 멀어지는 형태로 나옴

→ 3월 말 경 개발 완료 예상



Tx Lens 미 부착으로 인한 좌우 방향 광량 저하 문제



Integer Ambiguity 문제

1. Sample #KT-1.x

미 구현 사항 – Sensor Calibration

- Fixed pattern noise (FPN) 보정 안 됨 → 고정형 노이즈 발생
→ 측정 거리 짧아짐
- Sensitivity 및 nonlinearity 보정 안 됨 → 반사율 등에 따른 거리 오차 발생
→ DC 성분 존재 시 정확도 하락
- Tx signal modeling 안 됨 → 거리 측정값의 nonlinearity 가 커짐

→ 3월 말 경 개발 완료 예상

미 구현 사항 – Camera Calibration

- 픽셀 별 방향 데이터 없음 → Depth data 의 3D point cloud화 불가

→ 4월 중 개발



Fixed Pattern Noise로 인한 Depth Estimation Error 예시

1. Sample #KT-1.x

미 구현 사항 – High Dynamic Range

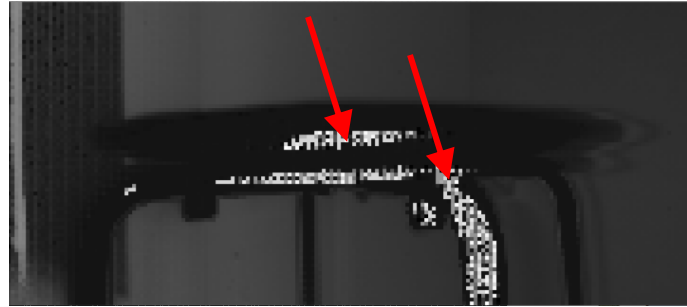
- 피사체가 가까이 있는 경우 픽셀이 saturation되어 측정 불가
- 현재 구현의 경우 밝은 물체(A4용지 등)의 경우 0.5m 정도 미만은 측정 불가
- Retroreflective material 측정 불가

→ 5월부터 개발

미 구현 사항 – Motion Blur Reduction

- 센서가 움직이거나 피사체가 움직이는 경우 blur 발생
- 현재 전반적으로 F/W가 naïve하게 구현되어 있어 발생하는 것으로 보임

→ 차후 최적화 진행 시 거의 해결될 것으로 예상됨



Saturation 발생



Motion Blur 발생(상), Motion Blur 없음(하)

1. Sample #KT-1.x

미 구현 사항 – Ethernet I/F

- 현재 UART 로만 구현되어 있는 상태
- 차후 센서 내부에 UART→Ethernet 모듈 추가하여 Ethernet 지원할 예정

→ 4월 말까지 개발

미 구현 사항 – Enclosure

- 현재 하이보 타입 (90도짜리) 개조하여 사용함
- 차후 KT 형태를 새로 제작할 예정

→ 4월 중순까지 3D print 혹은 CNC 형태로 샘플 제작

→ 4월 말~5월 초까지 사출금형 제작

1. Sample #KT-1.x

미 구현 사항 – Synchronization

- 현재 센서 간섭 문제 해결 안 됨

→ 4월 말까지 ID 지정하여 하는 방식으로 개발

2. Sample #KT-1.x 시험 환경 구축 – S/W

시험 환경

- 일반 PC (x86, Windows 환경)
- Visual Studio 2010 이상, C/C++ 환경
 - 2017, 2019로 마이그레이션해서 오른쪽과 같이 문제 해결하면 잘 동작하는 것 확인
- OpenCV 2.4.0 (프로젝트에 미포함)
 - 필요한 경우 아래 링크에서 받아서 설치 가능
- Pthreads-win32 2.8.0 (프로젝트에 포함)

Hybo 개발 PC

- Visual Studio 2010 사용
- System32, syswow64에 opencv dll 설치됨
- System32, syswow64에 thread building block dll 설치됨
- C:\opencv2.4에 opencv 2.4.0 윈도우 기본 빌드된 버전 설치됨
 - <http://gofile.me/6s4Dr/q1qFLjB9q>
 - 압축파일 내부 txt 대로 진행 (dll 넣는 부분만 하면 됨 – 아래쪽에 tbb 도 해야 함)
 - 2012년에 만들어둔 것이라 약간 차이 있을 수 있음
- Windows 10 64b
- i9-9900KF stock / DDR4 16GBx4@3400MHz

주의사항

- 다른 VS 버전으로 프로젝트 migration 시 include 등 프로젝트 설정 망가지는 경우가 있음 → windows.h 등 기본 헤더파일이 없다고 나옴
 - 기본값과 상이하다 보니 migration 동작에서 제대로 처리하지 못하는 것으로 파악됨
 - <https://stackoverflow.com/questions/43410631/visual-studio-2017-cant-find-windows-h> 참조 (본문에서 오른쪽 부분만 보면 됨)

This happens when you have customized include/library paths in legacy projects. If you added your own additional paths in project properties, VisualStudio 2017 can't automatically figure out base paths when switching between platforms/toolsets - normally it automatically puts correct paths there, but if you added customizations, VS won't touch them.

This is legitimate problem which I ran into myself recently when migrating old project targeted for Windows XP into VS2017. None of the answers or comments listed/linked here so far are helpful. I have all legacy SDKs in VisualStudio 2017 installer, and none of that fixed VS not finding essential includes such as `<windows.h>`. In my case the project was using v120 toolset from VS2013, which is superseded by v140_xp in newer VS.

After setting correct platform and toolset understood by VS2017, I did the following to resolve the problem:

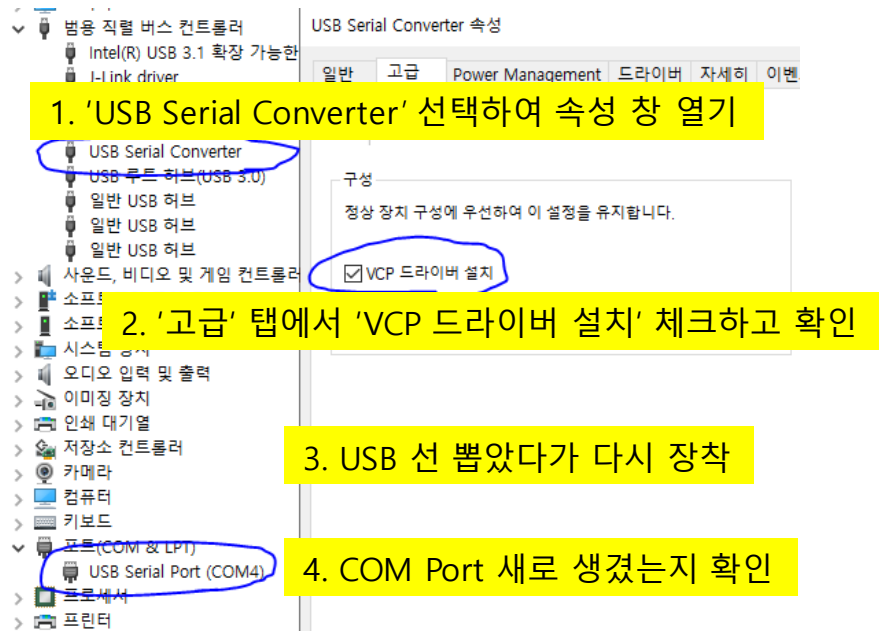
- Open project properties, go to VC++ Directories, for 'Include Directories' and for 'Library Directories', choose `<Inherit from parent or project defaults>`. This will remove your additional paths.
- Click 'Apply'. This will reset include path to something like `$(VC_IncludePath_x86);$(WindowsSdk_71A_IncludePath_x86)` (will vary for SDKs).
- Re-add your extra paths here, or better yet - under C/C++/General -> Additional Include Directories and Linker/General -> Additional Library Directories.

VS 버전 간 Migration 했는데 기본 헤더(windows.h 등) 없다고 빌드 안 되는 경우 해결 방법

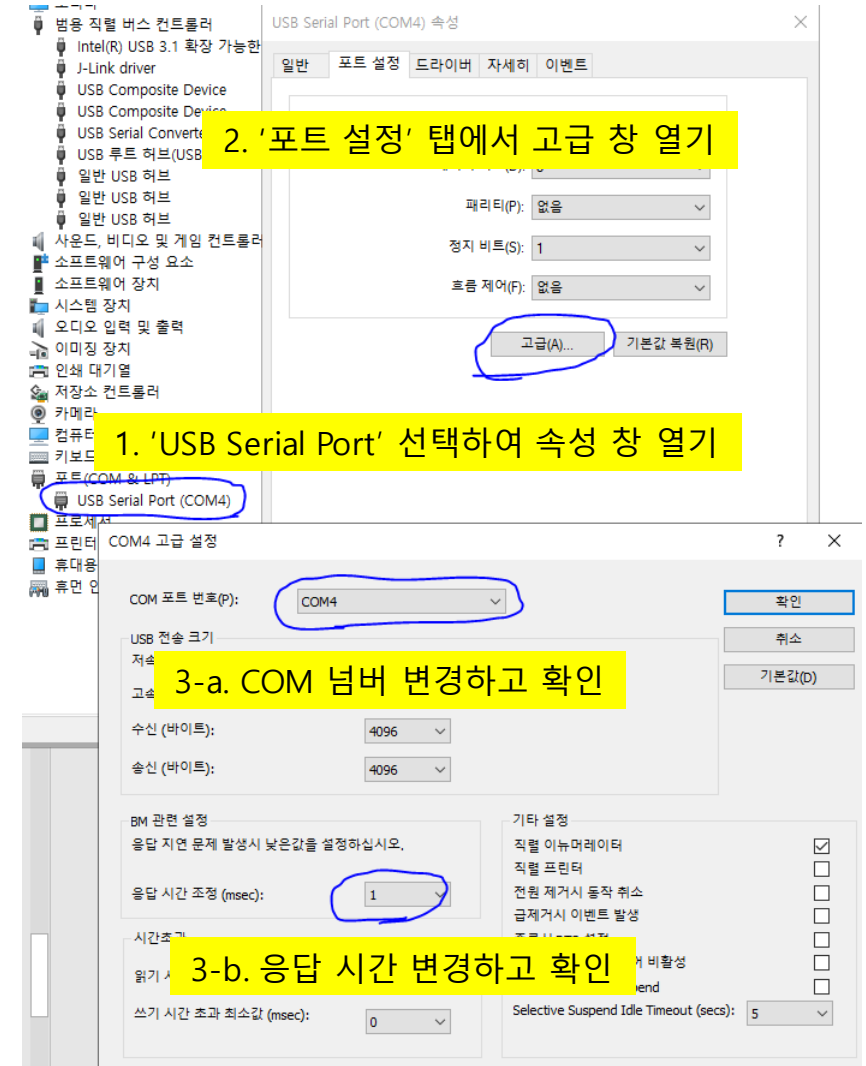
2. Sample #KT-1.x 시험 환경 구축 - H/W

FT232H 설치

- FTDI VCP 드라이버로 설치해야 함
 1. 센서에서 나오는 USB 선을 PC에 장착하여 인식되는지 확인 (인식 안되면 FTDI webpage에서 드라이버 다운로드받아 설치)
 2. 인식되면 장치 관리자에서 아래 그림과 같이 설정
- 이게 USB 포트별로 세팅이 저장되는 것으로 보이므로, 다른 USB 포트 사용시에는 할 때 마다 해 주어야 함
- COM 포트 넘버가 10을 넘어가는 경우 오류가 발생하는 경우가 많으므로 오른쪽 그림을 참고하여 가급적 9 이하로 설정
- 이미지 데이터가 깨지는 경우가 빈번하다면 오른쪽 그림을 참조하여 응답 시간을 1msec로 수정



FTDI VCP 드라이버로 설치하기



COM 포트 설정하기

2. Sample #KT-1.x 시험 환경 구축 – H/W

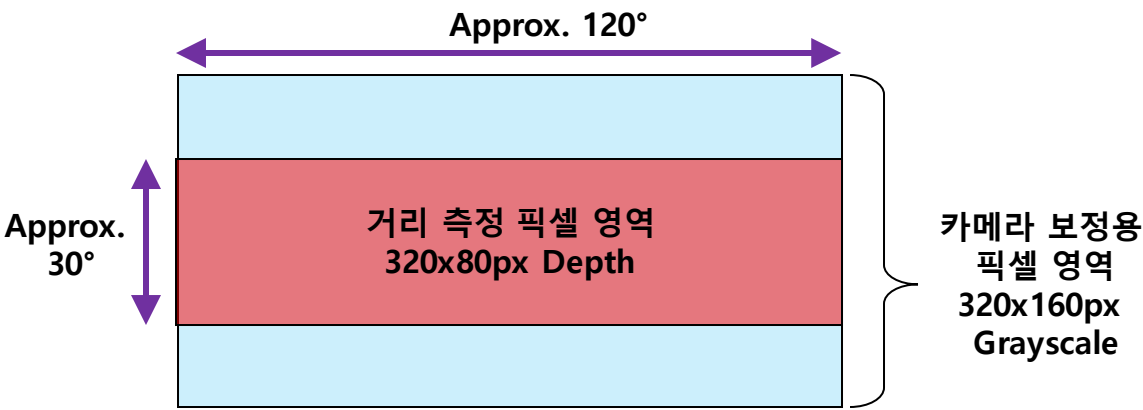
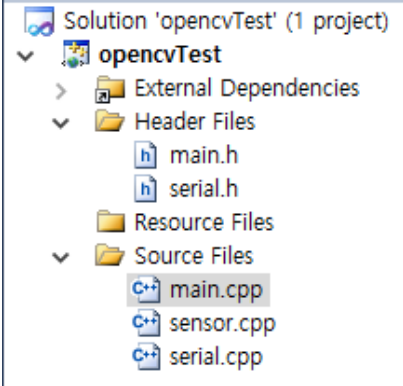
센서 전원

- 센서 전원은 센서 PCB 상에 11V 이상, 13V 미만으로 공급되어야 함
→ 센서 외부 전원 커넥터부터 PCB까지 가는 전선에서의 전압 강하를 고려하여 커넥터에서는 11.5V~12.5V 정도로 입력하는 것을 권장
- 입력 전원은 전압 regulation이 되어 있어야 함
- 내경 2mm 짜리 DC 잭 사용
- 동봉한 12V 1A 어댑터를 사용하여 220V 전원에서 전원 공급 가능
- 기타 12V, 0.5A(최대) 공급 가능한 다른 전원 사용 가능
- USB, 전원 켜는 순서는 상관 없음
- 정전기 주의

3. Sample S/W 간단 설명

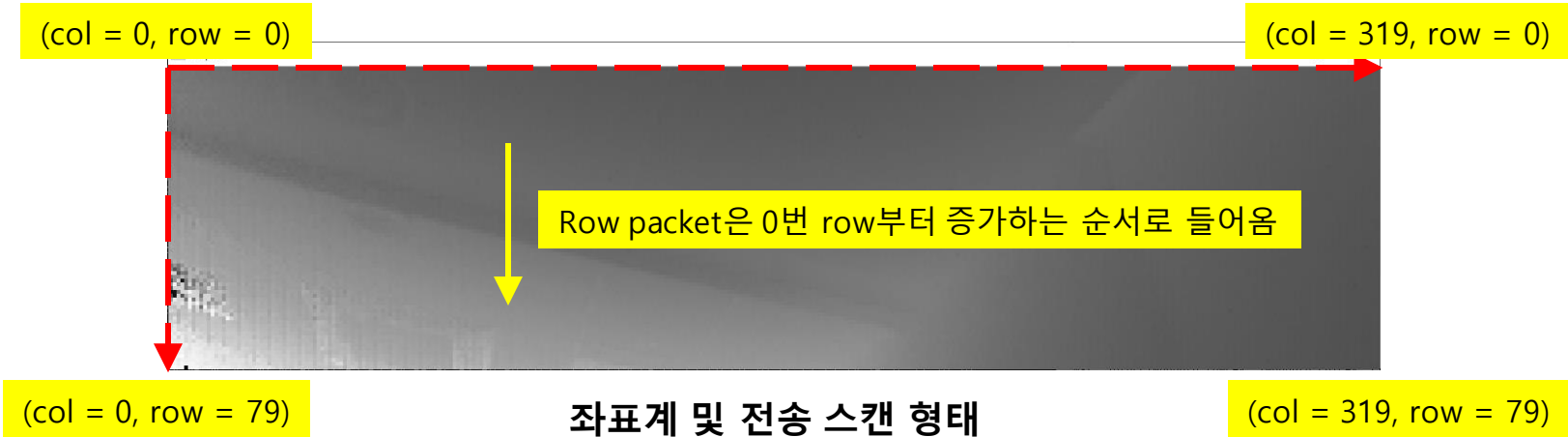
프로젝트 구조

- main.h: 쓰레드 선언 등
- serial.h, serial.cpp: 시리얼 포트 관련 코드
- main.cpp: 통신 쓰레드와 뷰어 쓰레드 구동
- **sensor.cpp: 통신 쓰레드와 뷰어 쓰레드 구현**



패킷 구조

- Row packet 정의와 row 전송 순서로 구성
- Row packet 정의
 - 아래 표 참조
- Row 전송 순서
 - 0번 row부터 증가하는 순서로 들어옴
- Gray → Depth → Gray ... 형태로 지속적으로 전송됨
- Gray 좌표계 → Depth 좌표계 = row 에 -40
- Depth 좌표계 → Gray 좌표계 = row 에 +40



Preamble	Row #	Image Type	Pixel 0	Pixel 1	...	Pixel 318	Pixel 319
0x5AA55AA5	0~79 (depth) 0~159 (gray)	0 (depth) 1 (gray)	0 ... 624 [cm] (depth) 0 ... 2047 [bits] (gray)				
4B	1B	1B	2B	2B	...	2B	2B

각 Row 패킷 구조

3. Sample S/W 간단 설명

- Thread kill 등은 따로 구현하지 않음 (Windows 에서 알아서 처리)
- 차후 Ethernet 구현도 동일한 구조로 작성
- 최대한 간결하게 작성

