|  |
| --- |
| **USB Host Driver** |
| 구조설계서 |
|  |
|  |
| **2022-01-14** |
| **김성후** |

이 문서는 **USB Host Driver** 개발을 위한 구조설계서이다.

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Description |
| 0.1 | 2022-01-14 | 김성후 | 초기 문서 생성 |
| 0.2 | 2022-01-16 | 김성후 | Interim 작성 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

[1. 시스템 개요 3](#_Toc479150191)

[2. 요구사항 4](#_Toc479150192)

[2.1. 기능적 요구사항 4](#_Toc479150193)

[2.2. 비기능적 요구사항 4](#_Toc479150194)

[2.3. 품질 속성 5](#_Toc479150195)

[3. 시스템 구조 6](#_Toc479150196)

[4. 컴포넌트 사양 7](#_Toc479150197)

[부록 8](#_Toc479150198)

[A. 도메인 모델 9](#_Toc479150199)

[B. 품질 시나리오 10](#_Toc479150200)

[C. 품질 시나리오 분석 11](#_Toc479150201)

[D. 후보 구조 12](#_Toc479150202)

[E. 후보 구조 평가 13](#_Toc479150203)

[F. 최종 구조 설계 14](#_Toc479150204)

[G. 최종 구조 평가(ATAM) 15](#_Toc479150205)

# 시스템 개요

* 1. **시스템 개요**

USB는 어떤 시스템이고 시스템 구성요소는 이러한 것들이 있는데(스펙 그림 첨부하고) 데이터를 어떻게 주고받고 열거하고 블라블라. 이러한 요소들이 필요하다. USB허브란 이런 것이며 이것을 개발할 것이다.



그림 - USB 허브 제품의 예(Belkin社 멀티포트 허브)

USB(Universal Serial Bus)1는 컴퓨터 본체(호스트)와 다른 주변 장치(디바이스)를 연결하는 입출력 표준 프로토콜의 하나이다. 그림 1. USB 연결 구조의 예 USB는 하나의 호스트(컴퓨터)와 다수의 디바이스를 연결한다(그림 1 참고). 디바이스는 한 개 이 상의 기능(Function)을 제공한다. 허브는 디바이스를 연결해서 USB 시스템에 포함시킬 수 있는 기 능이다. 호스트는 기본적으로 허브 기능을 포함하며 이를 루트 허브라고 한다. 그림 1에서 키보드 (KBD)는 키보드 기능 뿐 아니라 허브 기능을 포함하고 있으며, 이에 펜과 마우스 장치를 연결할 1 https://ko.wikipedia.org/wiki/USB 페이지 5 of 50 수 있다. 그림 2. USB 시스템 구조(USB 2.02 ) 호스트의 Application(Client SW)에서 연결된 디바이스가 제공하는 기능을 사용할 수 있도록 하기 위한 시스템이 USB Host System이다. USB Host System은 USB의 기본적인 통신을 담당하는 Host Controller(hardware)와 디바이스의 연결을 관리하고 통신을 중계하는 Host Driver(software)로 구 성된다.

* 1. **사업 환경**

USB 장치가 대중적으로 000억개의 장치가 사용될 정도로 활성화되어 있으나 기본적인 피씨에 탑재된 슬롯은 이를 모두 감당하기 어려우며, 더욱이 피씨는 본체 후방 슬롯을 사용하면 편의성이 떨어지고, 요즘 많이 사용하는 노트북은 데스크탑에 비해 상대적으로 슬롯이 적다. 그래서 USB 허브 제품이 많이 사용되고 있으며, 특히 많이 사용되는 맥북 계열의 제품은 최근 나온 뭐버전 외엔 C타입밖%에 ㅇ없어서 어쩌고

▪ 시스템의 동작 및 사업 환경은 각자 설정한다. – 어떤 기능 및 품질 측면에서 장점이 있는지 등.

* + 1. **Stakeholder**

|  |  |
| --- | --- |
| Stakeholder | Interest |
| OperationSystem 개발사 | 불특정한 종류의 다양항 USB Device에 대해서 최대한 정보를 파악할 수 있길 바라며, 그에 따른 최적의 데이터 전송을 기대한다. |
| Usb Device 개발사 | 개발한 장치가 타겟 운영체제에서 개발의도대로 동작하기를 기대한다. |
| LED Controller 개발사 | 다양한 제품에 범용적으로 사용될 수 있는 LED 제품을 개발하고 납품하고 싶어한다. |
| User | 가능한 많은 장치를 연결하여 사용하고 싶어한다. 합리적인 가격에 안정적인 속도와 충전 기능을 제공받고 싶어한다. |
|  |  |

* + 1. **시스템 제약사항**
* 하드웨어는 호스트와 연결부는 C타입과 USB타입의 단자 두 가지 버전이 제공되며 하드웨어적인 통신은 Usb Host Controller 장치가 담당한다.
* 수용할 수 있는 운영체제가 있는 클라이언트를 가정한다.
* 127개, 30미터
* 서브 허브까지 최대한의 제품이 원활하게 사용되기 위하여 허브에 자체 전원 공급장치가 있다.
* USB 2.0 버전을 지원하는 장치를 가정한다. 차기 제품에서 상위 버전에 대한 하드웨어 지원을 할 수 있으며 그에 따른 펌웨어 업데이트가 가능하다.
* ▪ 과제의 제약 사항에 대하여, – 솔루션이 최적화된 것을 판단할 수 있는 품질 시나리오가 검토되어야 한다. – 제약 사항의 변경에 대해서도 검토되어야 한다
* 외부 LED는 6개로 이루어져 있으며, Host와의 연결과 Hub와 연결된 5개의 Device를 표현하기 위한 장치이다. 파란색 불빛을 낼 수 있으며 점멸이 가능하다.
  1. **시스템 정의**

본 과제에서 개발하는 USB Host Driver의 개요 및 개발 범위는 그림2(변경)와 같다.

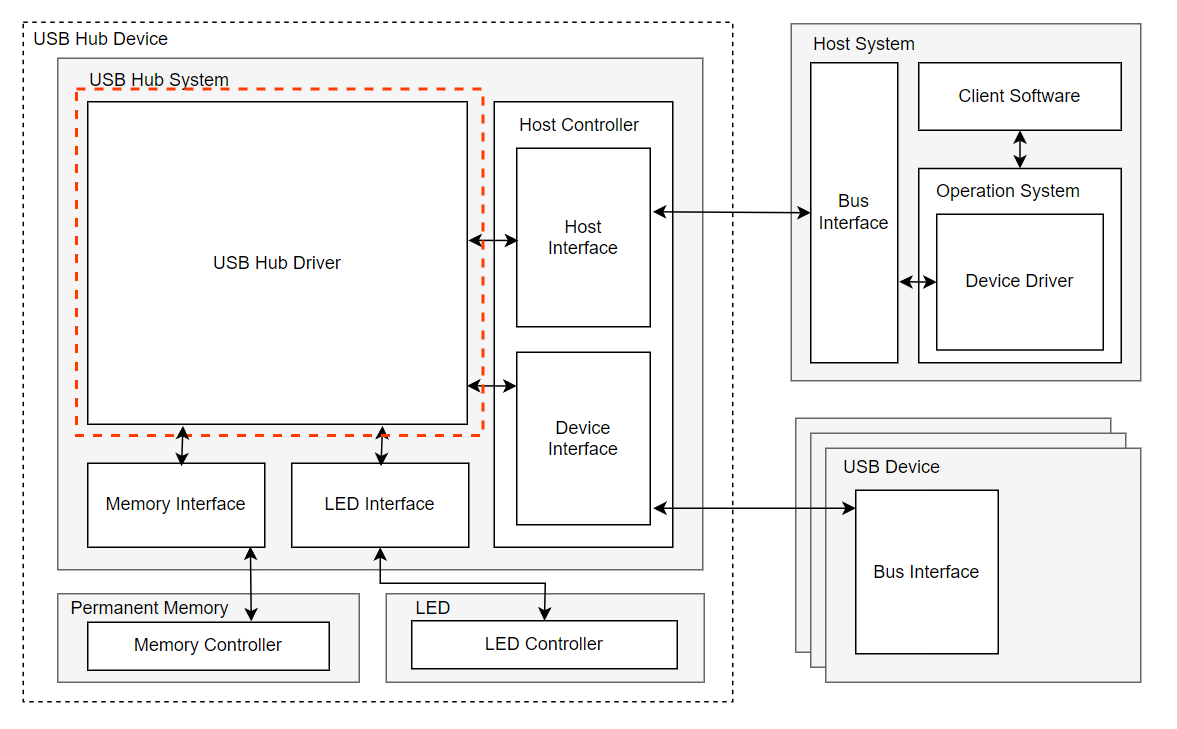


그림 - 시스템 전체 구조와 개발 범위

Usb specification에서 정의하는 물리적 연결 계층인 Host Controller에 대해서 Host Interface와 Device Interface로 기능을 분리하였다. 본 시스템이 Host 시스템에 embed된 장치가 아닌 호스트와의 연결도 가져가는 시스템인 점에서 착안하였다.

* + 1. **External Interface**

|  |  |
| --- | --- |
| External Interface | Description |
| Host Interface | 호스트 시스템과 통신할 수 있는 인터페이스이다. 데이터를 주고 받을 수 있는 두 개의 버퍼와 제어 정보를 주고 받을 수 있는 인터럽트 회선이 있다. |
| Device Interface | USB Device와 통신할 수 있는 인터페이스이다. Usb Hub Device의 구성요소로서 USB Host Driver에 의해 직접 접근이 가능하다. |
| Led Interface | Third-party 하드웨어인 LED Device를 제어할 수 있는 인터페이스이다. |
| Memory Interface | 시스템에 전원이 들어오지 않아도 정보가 저장되는 영구적인 메모리를 제어할 수 있는 인터페이스이다. |

* + 1. **시스템 경계**

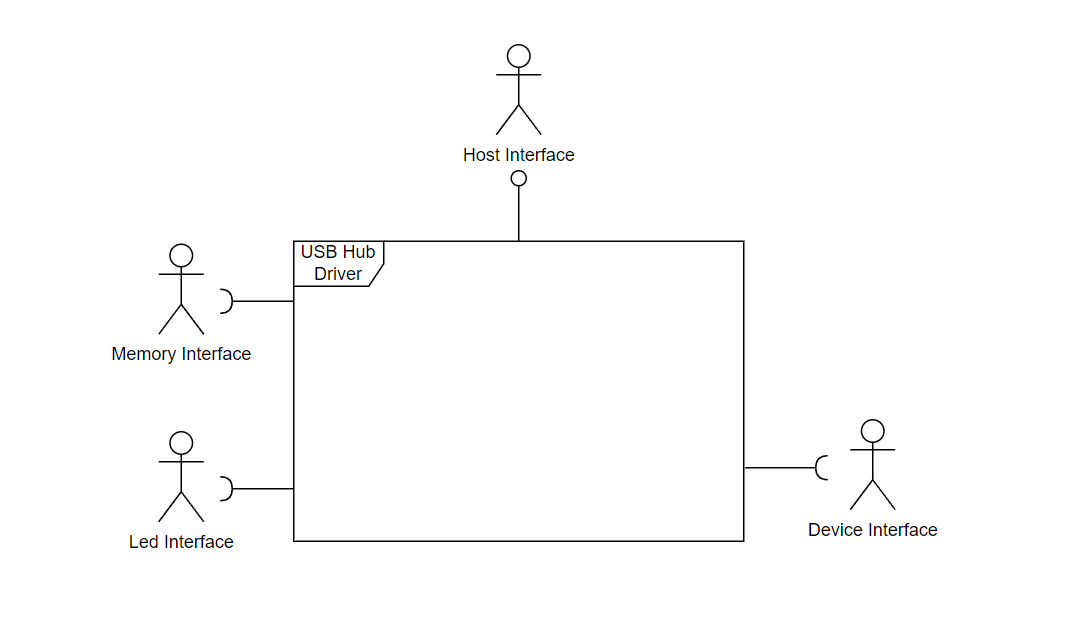


그림 - 시스템 경계

Host Interface와 Device Interface는 각각 연결된 장치로부터 데이터 송수신을 요청할 수 있다. 또한 Device Interface는 장치의 상태를 통보할 수 있으며, Host Interface는 장치의 상태를 요구하고 제어를 시스템에게 요청할 수 있다. 시스템은 외부 LED 하드웨어에 대한 제어를 Led Interface에게 요청할 수 있다.

* + 1. **시스템 동작**

// 점검1-1. 시스템의 경계(정의)가 명확한가?

// 점검1-2. 시스템의 동작/사업 환경에 대한 설명이 충분한가?

# 요구사항

## 기능적 요구사항

* + 1. **유즈케이스 다이어그램**

시스템 경계에 따른 유즈케이스 다이어그램은 아래 그림4(변경가능)과 같다.

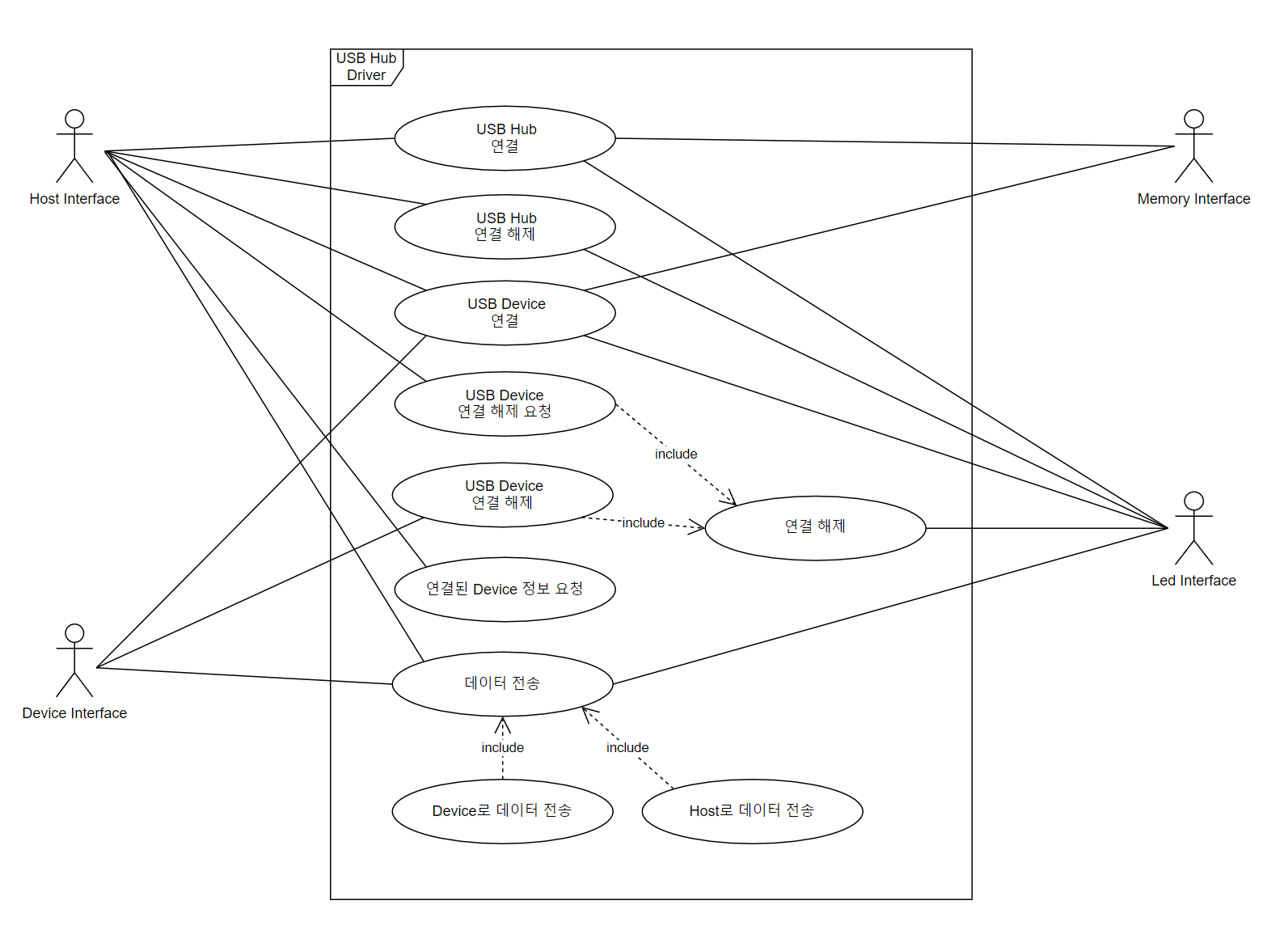


그림 - 유즈케이스 다이어그램

디바이스 정보 요청은 시스템이 디바이스와 연결시 컨텍스트를 가지고 있으므로 디바이스가 참여하지 않는다.

// 활동2. 기능 명세

// 점검2-1. 구조에 영향을 미치는 기능 명세가 충분한가?

// 점검2-2. Use Case의 관계가 명확한가? (include/extend 등)

// 점검2-3. 기능의 구분이 명시적인가? (stereotype 사용)

|  |  |
| --- | --- |
| **UC\_01** |  |
| 설명 |  |
| 행위자 |  |
| 선행조건 |  |
| 후행조건 |  |
| 기본 동작 |  |
| 추가 동작 |  |

기본적인 기능 – USB 디바이스의 연결을 관리한다. – USB 디바이스의 제어 및 데이터 전송을 관리한다. – USB 디바이스의 전원을 관리한다. ▪ 부가적인 기능 – Mass Storage Class

등시성과 아닌걸로 두개 구분? => 부가 요구사항에 대용량장치가 있는데 특별한 특성은 없고 평소처럼 전송 완결성을 중요시

인터럽트랑 또 뭐시기는?=>설계중 구성요소러서

## 비기능적 요구사항

// 활동5. 품질 속성 선정

// 점검5-2. 품질 요구사항의 명세가 적절한가?

// 점검5-3. 품질 요구사항의 측정이 가능한가?

성능 – 디바이스의 연결, 제어 등의 성능이 빠를 수록 좋다. ▪ 변경 용이성 / 확장성 – 하드웨어 사양의 변경이 용이할 수록 좋다. – 다양한 클래스로의 확장이 용이할 수록 좋다

* 뎁스 7 이하
* 총 연결 개수 127개 이하
* 펌웨어 업데이트를 통해 클래스를 추가할 수 있어야 한다.
* 전원 차단시 데이터 보호 => 클라이언트 소프트웨어에 영향 안가도록
* 한번 연결된 장치는 더 빨ㄹ ㅣ연결
* 여러 디바이스에 독립성 제공(스트림 파이프 별개)

// 점검5-4. 비기능적 요구사항의 제약이 명확한가?

|  |  |  |
| --- | --- | --- |
| **NFR\_01** |  |  |
| 설명 |  | |
| 환경 |  | |
| 자극 |  | |
| 반응 |  | |
| 측정 |  | |
| **제약** |  | |

## 품질 속성

// 활동5. 품질 속성 선정

// 점검5-2. 품질 요구사항의 명세가 적절한가?

// 점검5-3. 품질 요구사항의 측정이 가능한가?

|  |  |  |
| --- | --- | --- |
| **QA\_01** |  |  |
| 설명 |  | |
| 환경 |  | |
| 자극 |  | |
| 반응 |  | |
| 측정 |  | |

* 디바이스가 요구하는 전력, 대역폭등은 베스트 이포트로 제공해야 한다.

(네고시에이터)

* - 유에스비 클래스에 적합한 전송속도로 제공해야 한다.
* => 시나리오 키보드 로우 스피드, 오디오 풀스피드, 비디오 하이스피드
* 영상과 오디오와 같이 품질보다 등시성이 중요한 경우 등시성을 지킬 수 있어야 한다.
* => 스트림 파이프 열 때 특성 지정
* 조작된 유에스비 클래스 인식가능? 배드유에스비인가 무시긴가 막을수있나
* 다른 버전에 대한 하위호환
* => 네고시에이터가 상하위 버전 디바이스 입력시 버전도 정해준다고 가정

부록

[A. 도메인 모델 9](#_Toc479150206)

[B. 품질 시나리오 10](#_Toc479150207)

[C. 품질 시나리오 분석 11](#_Toc479150208)

[D. 후보 구조 12](#_Toc479150209)

[E. 후보 구조 평가 13](#_Toc479150210)

[F. 최종 구조 설계 14](#_Toc479150211)

[G. 최종 구조 평가(ATAM) 15](#_Toc479150212)

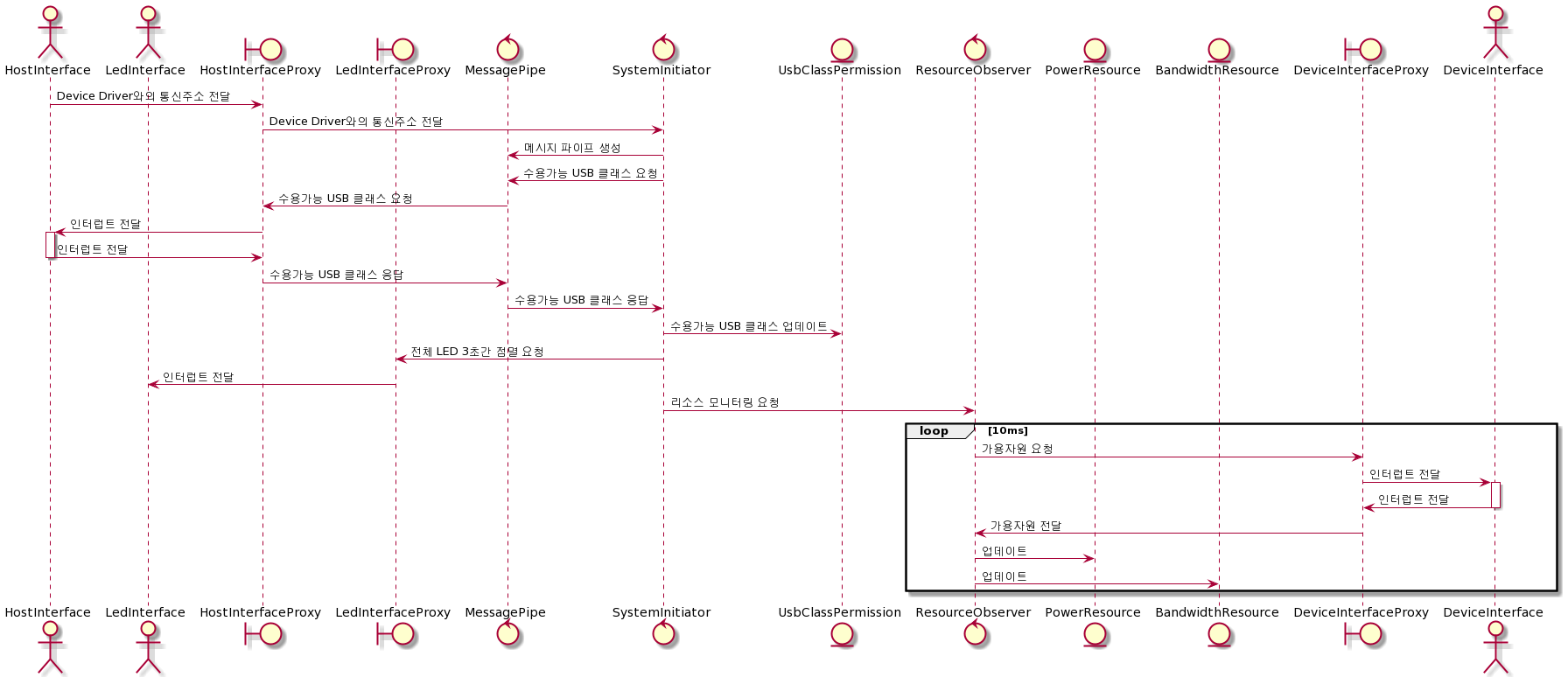
1. 도메인 모델

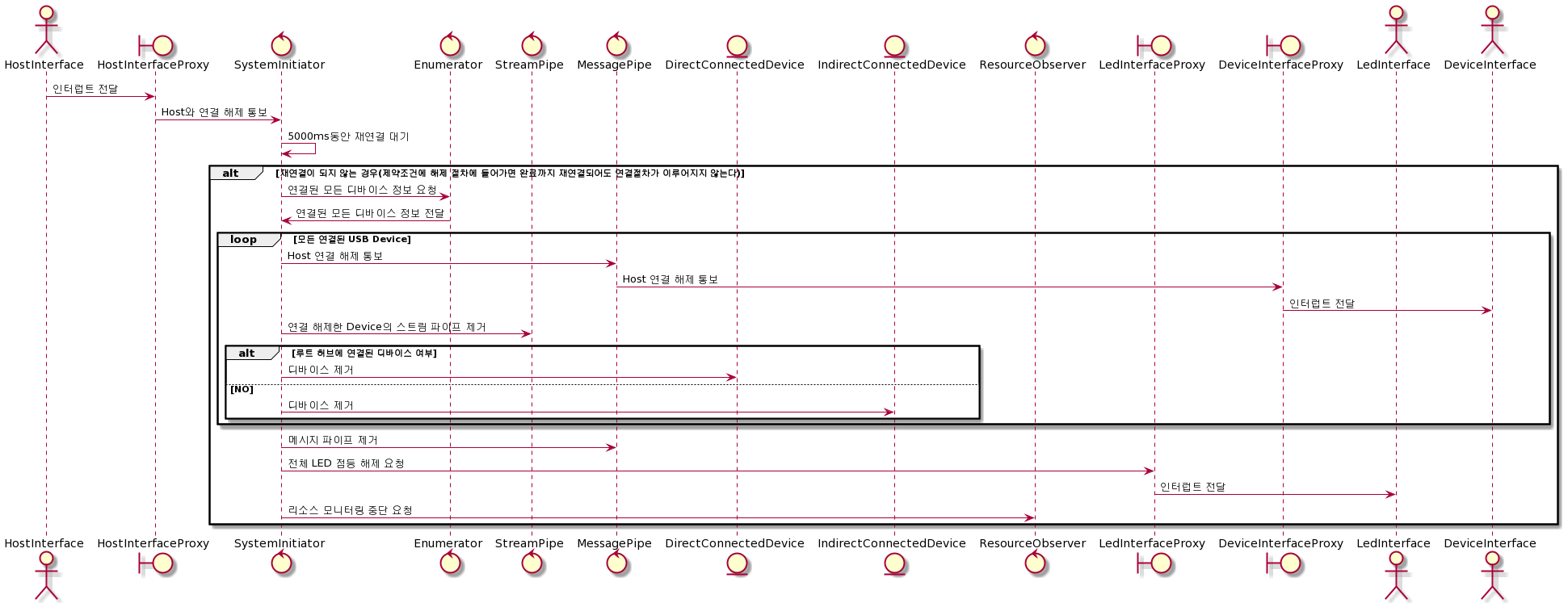


// 활동3. 도메인 모델 정립하기

// 점검3-1. 도메인 모델이 충분히 세분화 되었는가?

// 점검3-2. 도메인 모델에 구조가 반영되지 않았는가?





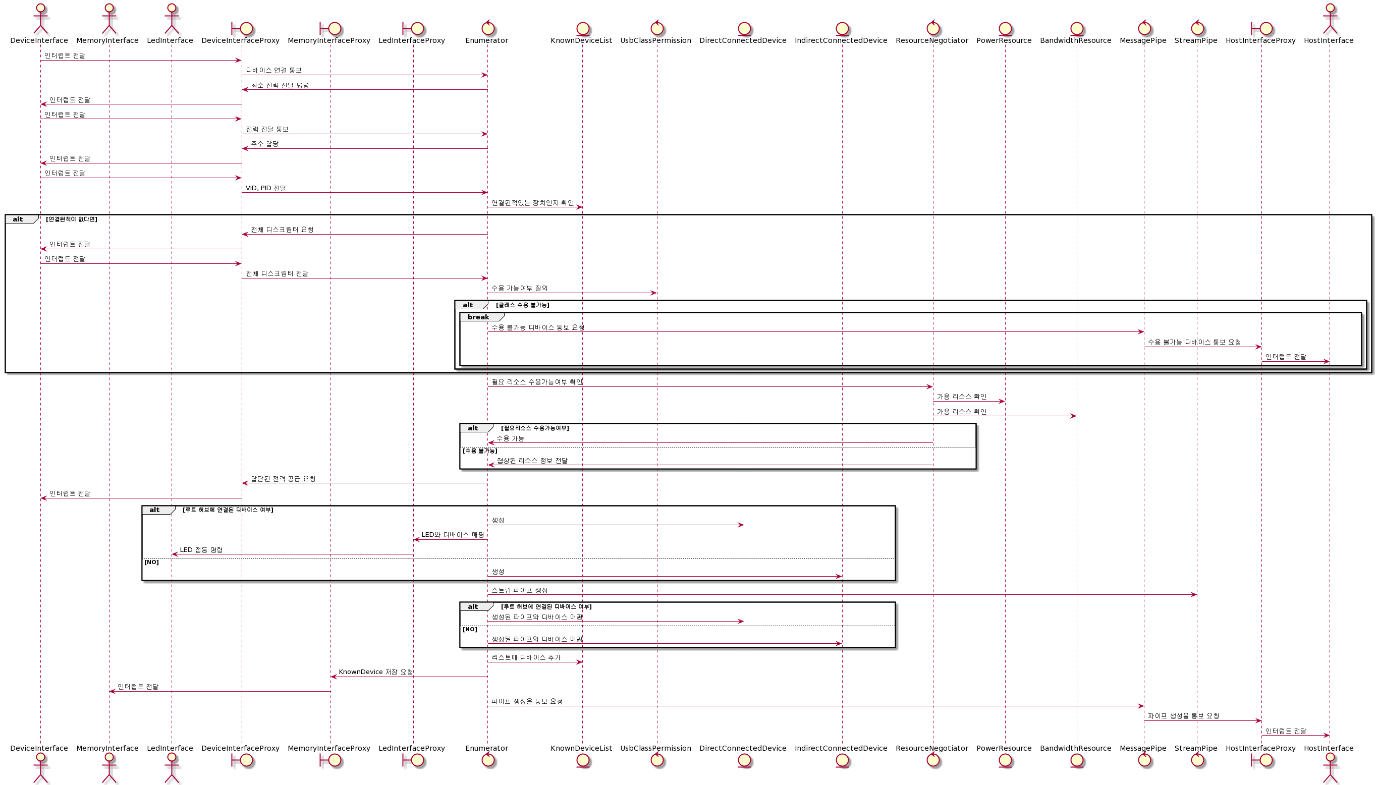
5초간의 재접속시간을 기다린다. 5초 이후에는 모든 리소스에 대핸 해제작업을 거치며 이동안은 새로 허브연결이 되어도 해제가 완료될때까지 응답하지 않는다. 다시 연결시 스펙에서 정의한 것처럼 재열거 과정이 이루어지게 된다.

* 허브 자체 전원 자단시 호스트로 오는 전원만 받을 수 있는데 이런 경우 다시 리소스를 설정하도록 알터 시나리오

USBDevice

* 장치를 추상화한 도메인으로 장치를 조작하기 위한 오퍼레이션이니 어드레스니 어쩌고 내부로 감춰서 직관적으로 USBDriver 도메인을 조작해서 어쩌고저쩌고
* 실제 정의된건 장치를 조작할 수 있는 api이지만 그걸 캡슐화하고 추상화해서 만들어

프록시 도메인들은 명령을 해석하고 변환하여 전달하는 역할을 한다.



타이니 유에스비 분석하면서 컴포넌트와 함수와 연결관계를 정리하고

거기에 스펙정리한거 하나씩 적용하면서 도메인모델부터 만들고 역으로 가자

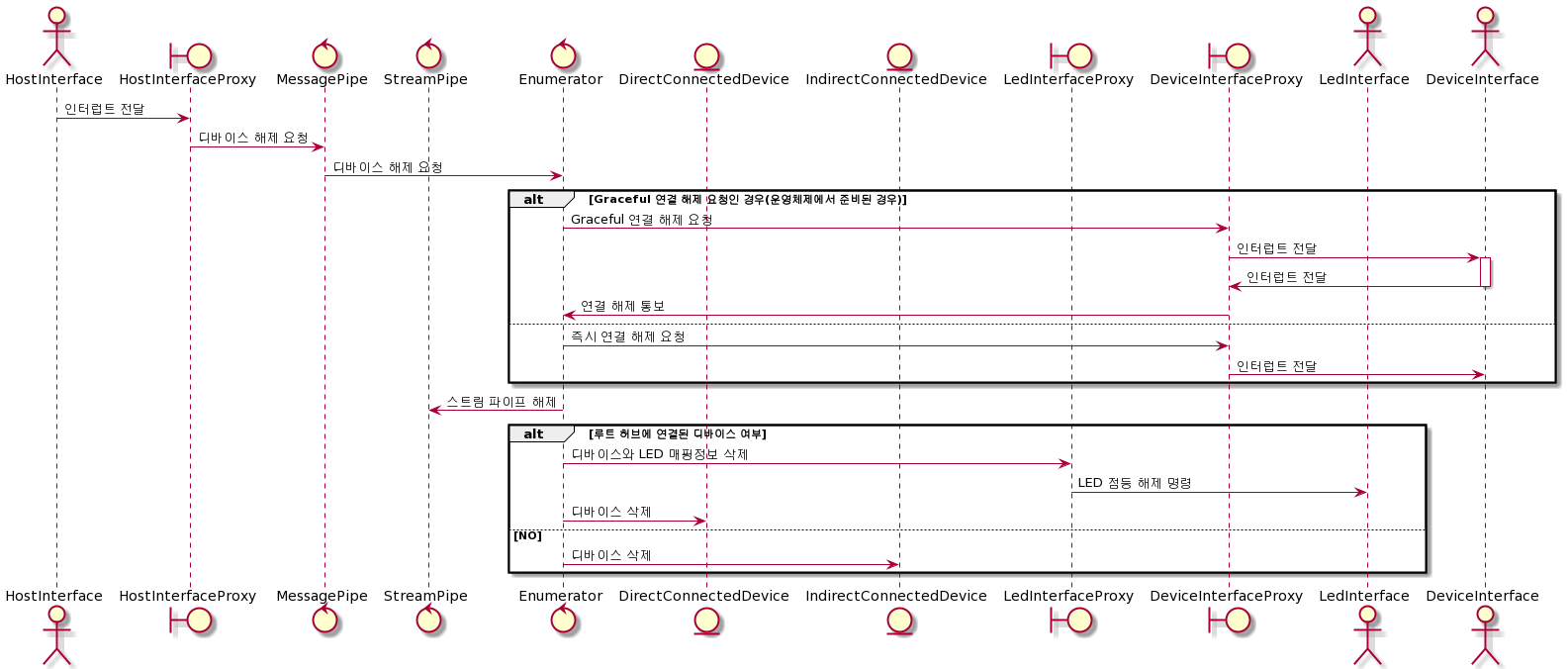
어쩌면 내일 코드보고 정리하고 내용 정리하는데 끝날수도 있다.

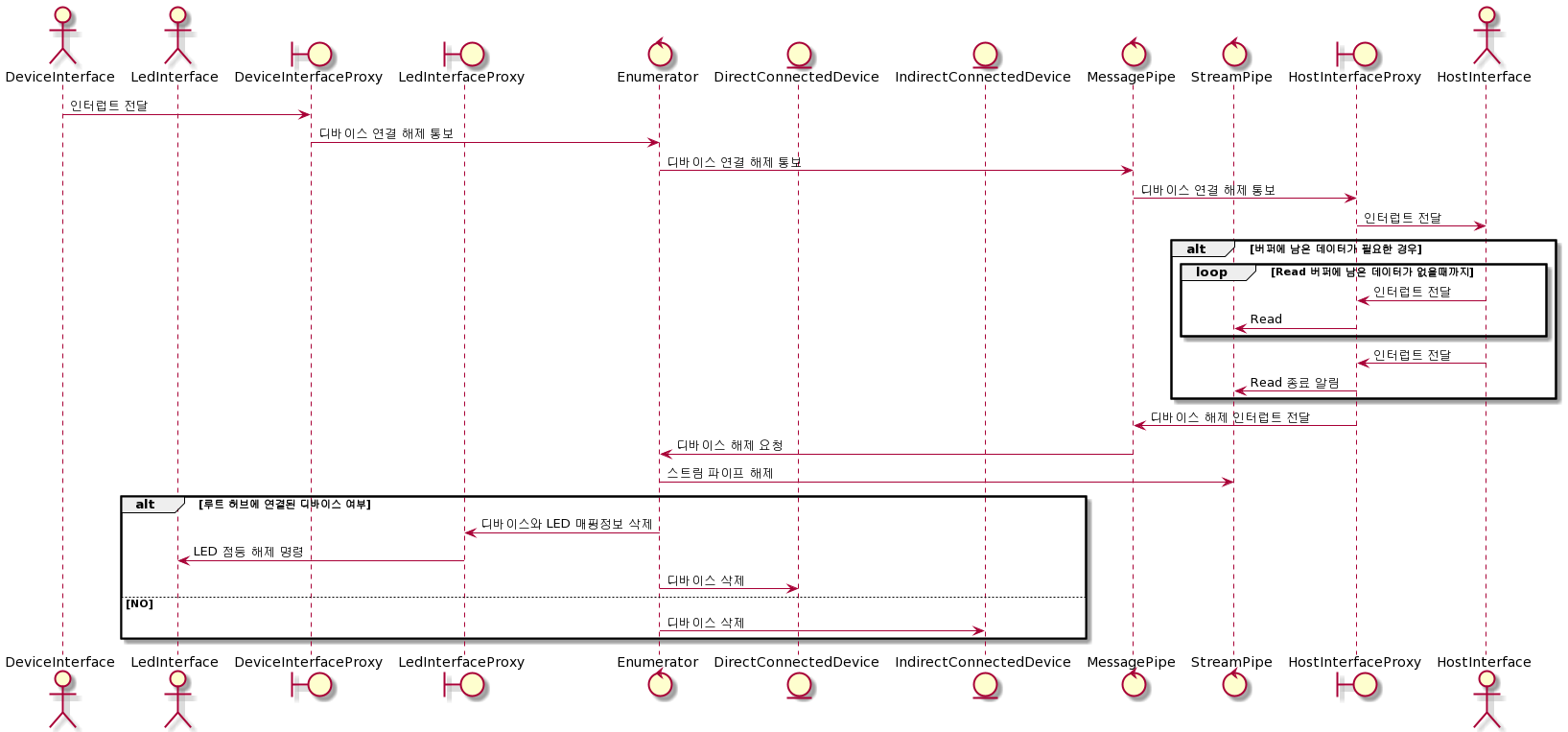
일요일에 많이 쓰면 되지 뭐…

맥북용 허브인데 허브뿐만 아니라 당연히 충전도 가능하고

그래서 그냥도 쓸수있고 자체 전원도 잇다.

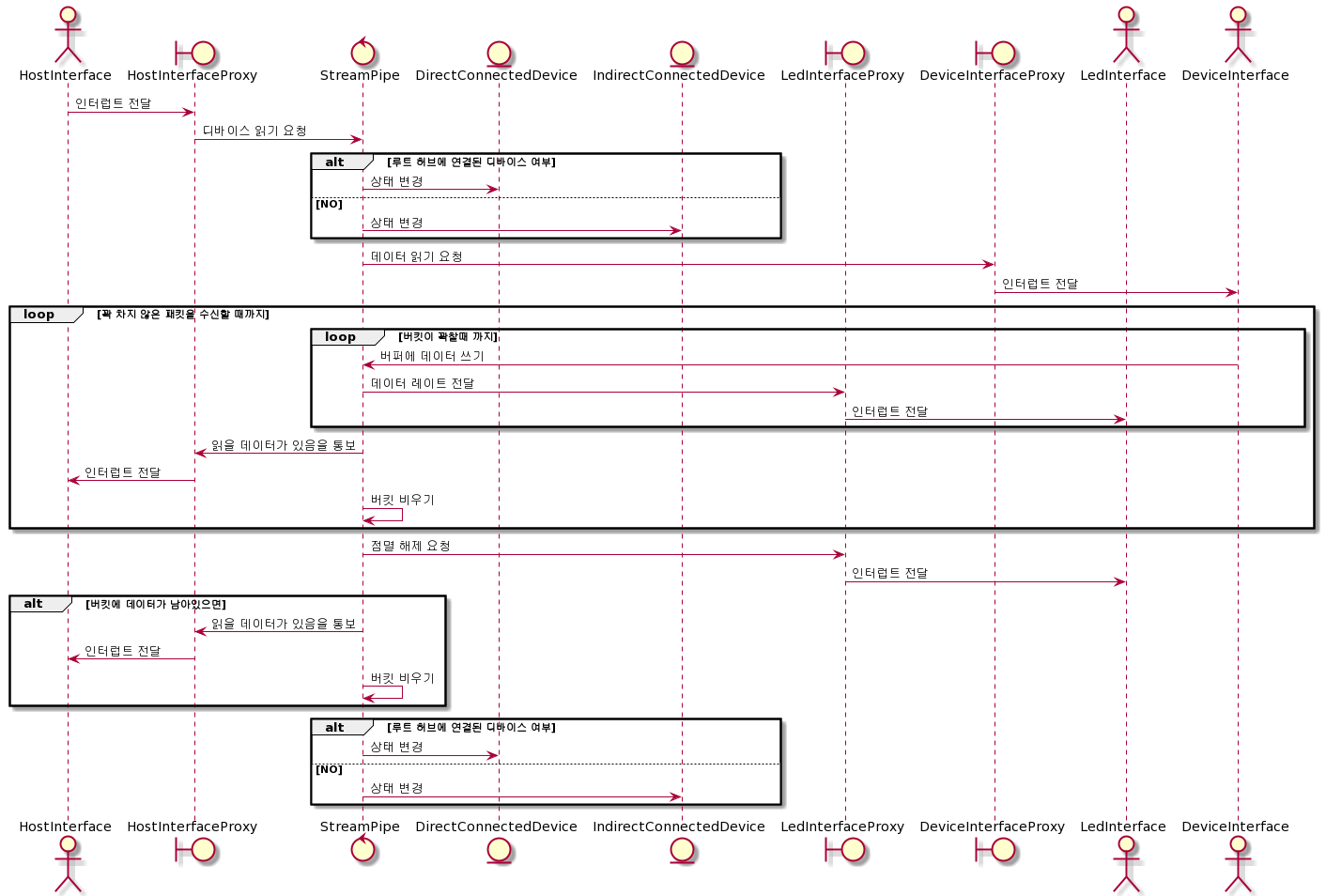
테스트 시나리오

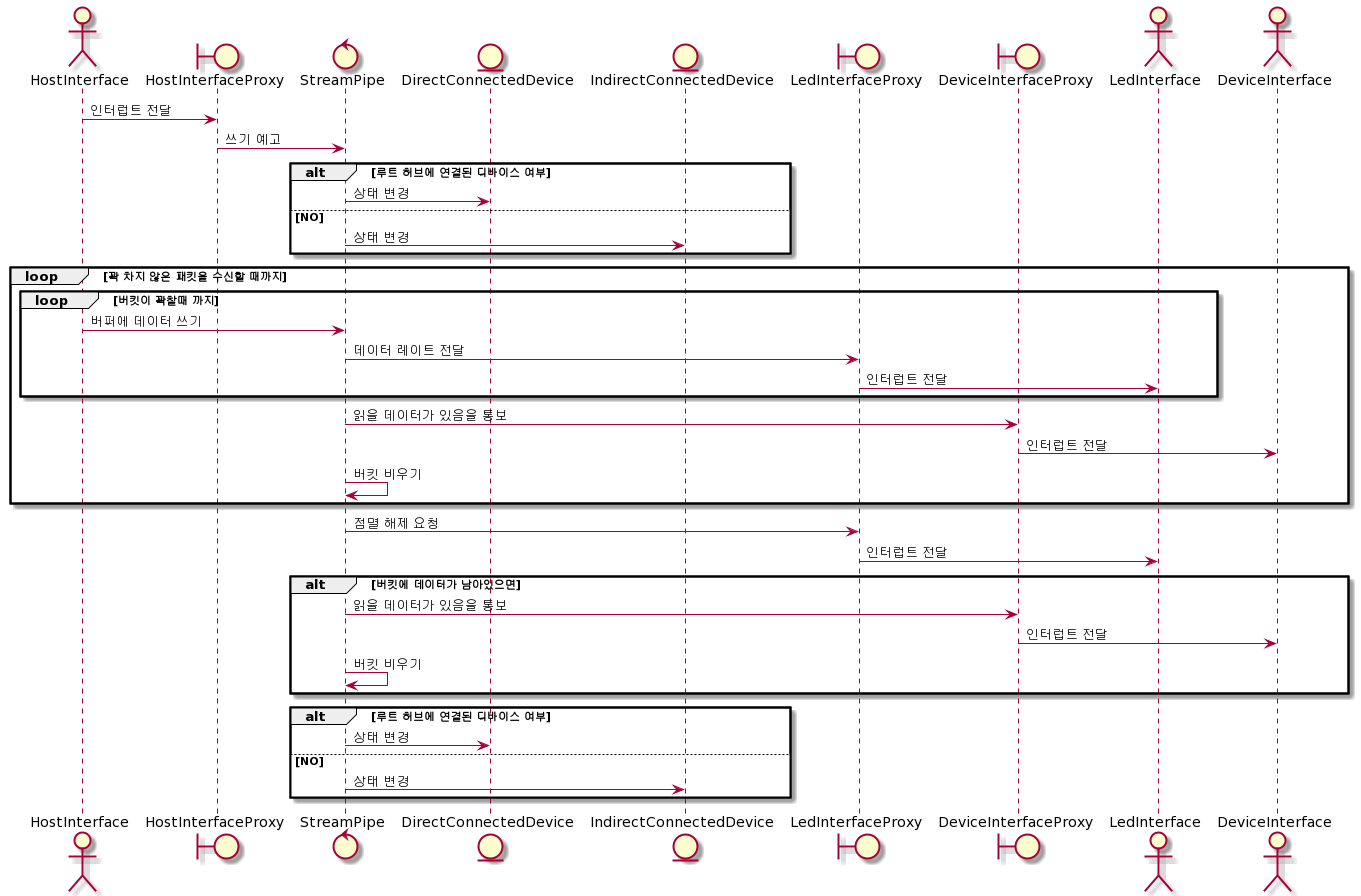


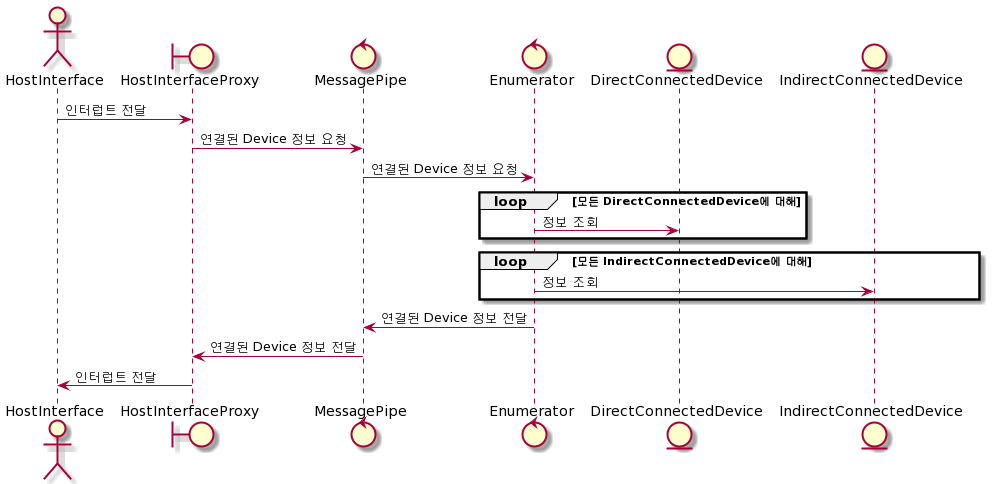


Enumerator -> MessagePipe: 디바이스 연결 해제 통보(데이터 손실에 주의할 수 있도록 하기 위함 => 비기능요구사항)

OperationSystem -> StreamPipe: Read(파이프 버퍼에 남은 데이터까지는 리드할 수 있도록 하기 위함 => 비기능 요구사항)







1. 품질 시나리오

// 활동4. 품질 시나리오 생성

// 점검4-1. 구조에 영향을 미치는 품질에 대한 검토가 충분한가?

// 점검4-2. 성능에 대한 검토가 충분한가?

// 점검4-3. 변경용이성/확장성에 대한 검토가 충분한가?

1. 품질 시나리오 분석

// 활동5. 품질 속성 선정

// 점검5-1. 품질 시나리오 분석이 적절한가? (근거)