
ubpulse H3 USB 개발가이드

ubpulse H3 USB HID 호스트 앱 개발 가이드.

Doc. ID. LXD146 V1

Release Date. 2017-05-23 .

Abstract - ubpulse H3 의 USB HID(Human Interface Device) 연결 된 호스트측(윈도우, 안드로이드, 리눅스등의 모든 운영체제)에서 ubpulse H3 와 통신하기 위한 USB HID 로레벨 통신부 개발 가이드.



목차

LAXTHA 기기의 USB HID 기반 통신.....	3
USB HID 기기의 USB 연결시 드라이버.....	4
USB HID 기기를 스마트폰(안드로이드 운영체제)에 연결하는 법.....	4
스마트폰에서 USB HID 기기 인식 상태보기.....	5
USB HID 기기를 윈도우 운영체제 PC 에 연결한 경우.....	7
USB HID 기기의 호스트측 통신 앱 개발 가이드.....	8
UBPULSE H3 의 호스트측 USB 수신버퍼 데이터 파싱.	9
LAXTHA USB 기기에서 USB HID 로레벨 버퍼로 데이터 전송하는 방식 이해.	10
호스트 측에서 USB HID 버퍼에서 데이터 읽는 방법.....	11
호스트측에서 USB HID 수신버퍼에서 실시간 데이터 읽기 처리시 주의사항.	12
UBPULSE H3 의 호스트 측 USB 송신버퍼.....	13
REVISION HISTORY	14

LAXTHA 기기의 USB HID 기반 통신.

USB HID (Human Interface Device) 클래스는 현존 모든 운영체제(windows, 리눅스, 안드로이드, iOS) 에서 지원되며 사용자가 별도의 드라이버 설치 하지 않아도 기기의 연결즉시 앱측에서 접근하여 사용가능한 상태가 달성된다. USB HID 가 도입된 LAXTHA 기기 역시 모든 운영체제에서 연결즉시 앱에서 접근가능한 수단이 운영체제 수준에서 지원된다. USB HID 를 통신매체로 사용하는 경우에도 통신시 여러 종류의 데이터표현이 되어야 하므로 LAXTHA USB HID 기기가 실제 통신시 송수신 하는 데이터 형식은 통신 패킷인 LXSDf T2/ LXSDf UH1 등의 범용 시리얼 통신 포맷이 사용된다.



...	7	6	5	4	3	2	1	0
	Ch1 Stream	PCD	PUD1	PC	PCDType,PUD2,CRD	PUD0	TS4	TS3

USB HID 기반 통신시 사용되는 데이터 포맷은 일반 시리얼 통신 규격인 LXSDf T2/ LXSDf UH1 등 이 가능함.

USB HID 기기의 USB 연결시 드라이버.

LAXTHA USB HID 기기는 HID(Human Interface Device) 클래스 표준을 준수하고 있다. USB HID 드라이버가 윈도우, 안드로이드, 리눅스 운영체제에서 기본적으로 제공되고 있기 때문에 별도의 드라이버 설치과정이 요구되지 않는다.

USB HID 기기를 스마트폰(안드로이드 운영체제)에 연결하는 법.

안드로이드 운영체제를 지원하는 스마트폰에 마이크로 USB 젠더를 장착하고 USB 케이블로 기기측의 USB 케이블과 연결한다.

연결예시. 아래 사진에서 “스마트폰에 연결” 부분처럼 커넥터 규격이 다르기 때문에 USB 젠더를 중간에 두고 있다.

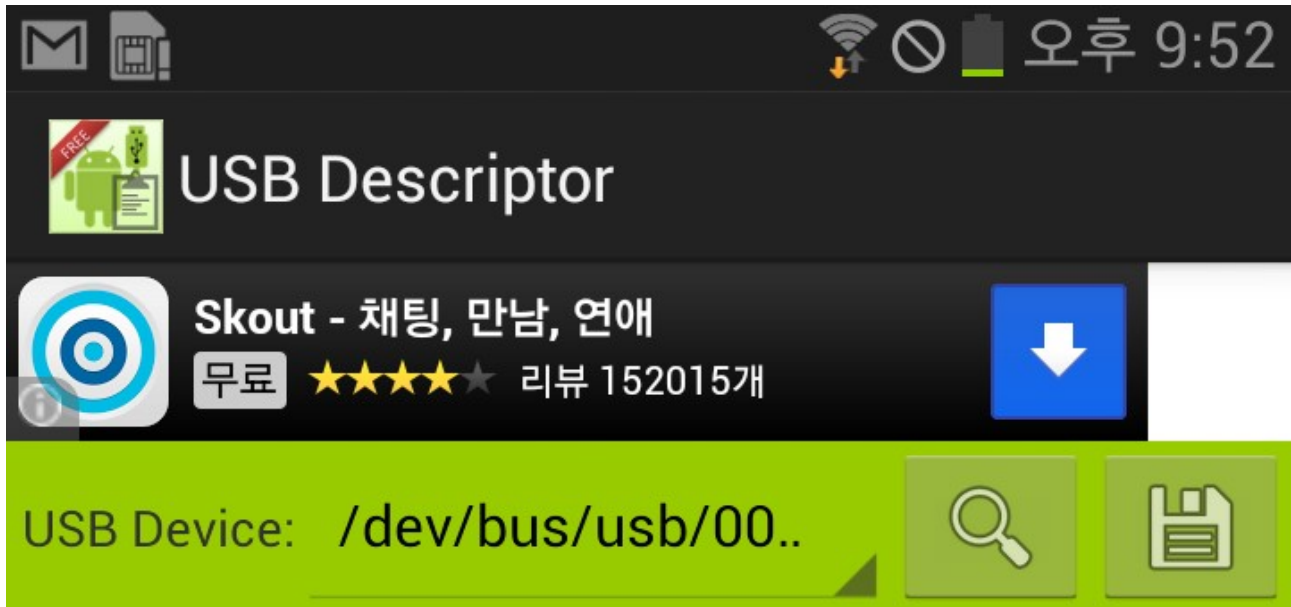


스마트폰에서 USB HID 기기 인식 상태보기.

일반 PC에서는 USB HID 장치 인식여부는 장치관리자에서 확인가능하다. 안드로이드 스마트폰에서는 기본 제공하는 앱이 없다. 별도로 무료배포되는 앱을 스토어에서 다운로드 받아서 실행해보면 USB HID 기기의 인식 상태를 확인가능하다. “USB Descriptor” 의 예를 보인다.



USB HID 기기를 스마트폰에 연결하고 인식된 상태를 보면 “USB Descriptor” 앱을 실행하면 LAXTHA 의 USB HID 기기가 안드로이드 운영체제에서 인식되어있음을 알 수 있다. 아래 그림.



Device ID: 2003

Device Name: /dev/bus/usb/002/003

Device Class: Unspecified(0x00)

Device Sub Class: Unknown(0x00)

Device Protocol: Unknown

벤더아이디 : 3871은 USB협회에 정식등록된 LAXTHA의 고유 아이디.

Vendor ID: 3871

Product ID: 65529

Interface Count: 1

Interface ID: 0

Interface Class: HID (Human Interface Device) 0x03

Interface Sub Class: No Subclass(0x00)

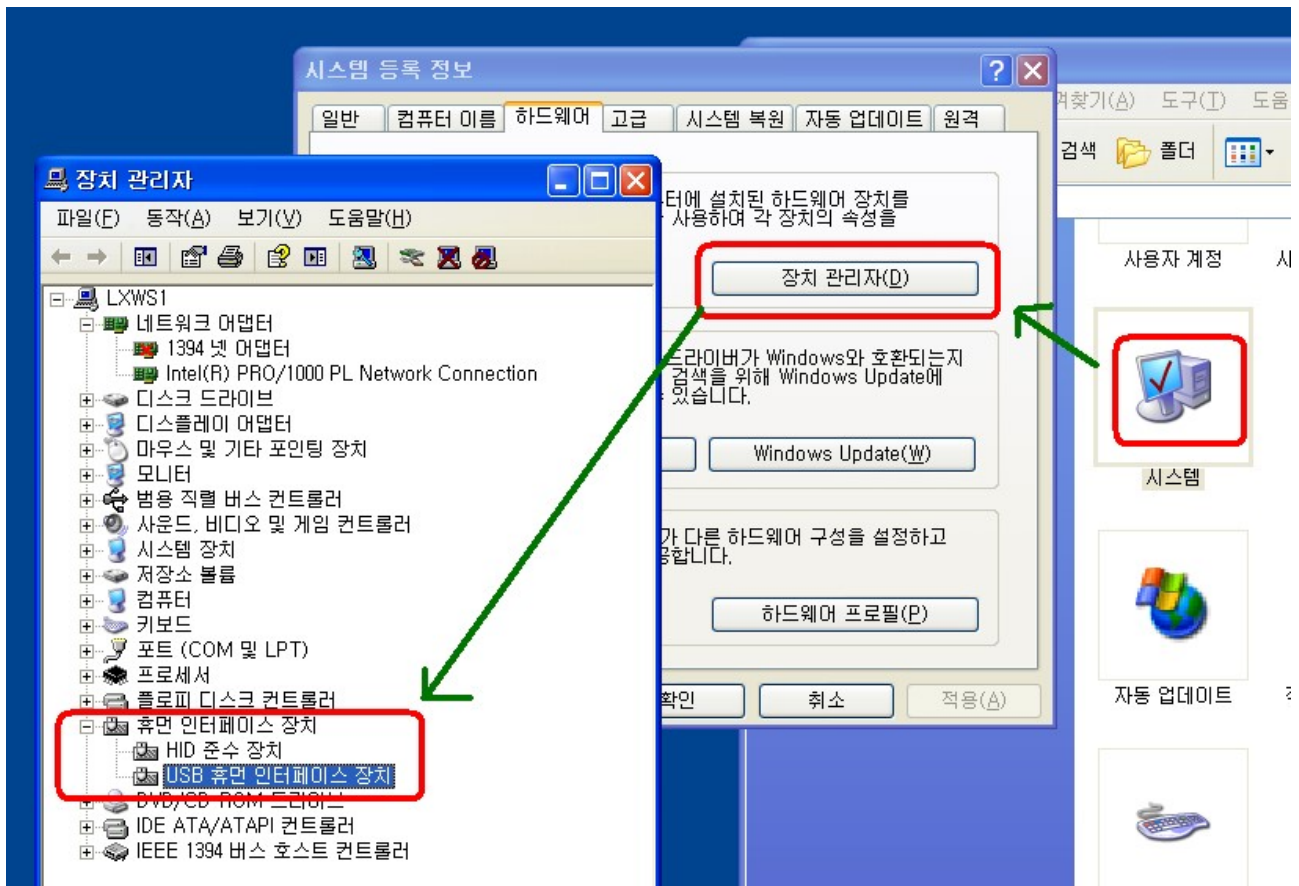
Interface Protocol: None(0x00)

Interface Endpoint Count: 2

USB HID로 인식하고 있음을 보임.

USB HID 기기를 윈도우 운영체제 PC 에 연결한 경우..

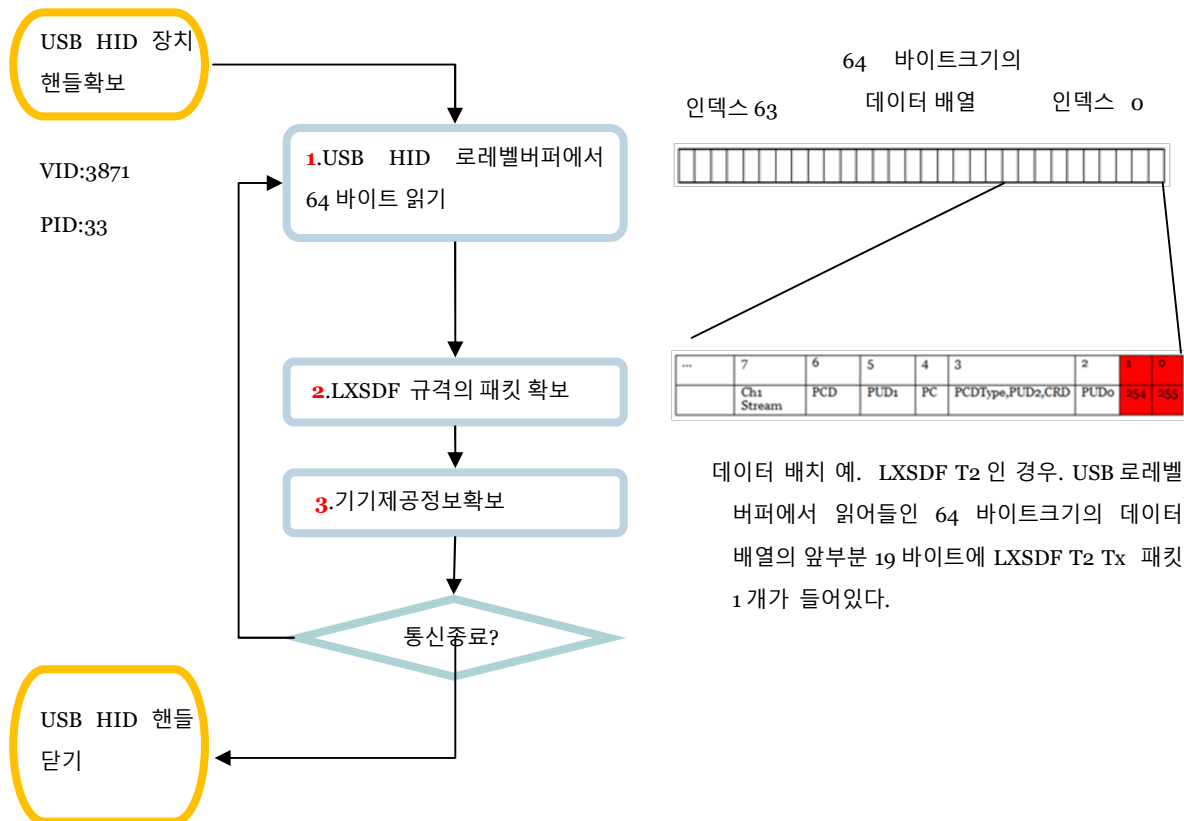
USB HID 기기를 윈도우 운영체제의 PC 에 USB 케이블로 연결하면 인식된 상태를 제어판 -> 시스템 -> 장치관리자에서 확인가능하다.



USB HID 기기의 호스트측 통신 앱 개발 가이드.

LAXTHA USB HID 기기 를 USB 호스트기기(스마트폰등)에 연결된 상태에서 앱에서 장치에 접근하고 장치에서 전송하는 데이터를 앱에서 프로그램 구현자에게 필요한 정보가 정리된다. 윈도우 운영체제의 Visual C++ 의 언어를 주로 사용하여 설명이 되고 있지만, 모든 프로그램 개발언어에서 해당하는 유사한 개념으로 대응시켜서 파악해야한다.

아래 그림이 앱에서 수행해야할 전체 처리 과정을 보이고 있다. 가장 먼저 호스트에 연결된 USB HID 의 장치 핸들을 잡는것으로 시작한다. 이때 사용되는 식별 정보로 usb vendor id = 3871 과 product ID 를 이용한다. Product ID 는 제품모델별로 다른 값이 할당 되어있으며, 예로 neuroNicle E1 인 경우 product ID 는 62(십진수)로 지정되어있고 ubpulse H3 는 33(십진수) 로 지정되어있다. USB 로레벨 버퍼(장치 인식과정중에 운영체제에서 자동으로 65 바이트크기의 버퍼를 생성한다.)에 접근하여 65 바이트를 한 번에 읽어온다. 첫번째 바이트는 버리고 2 번째부터 총 64 바이트크기의 데이터를 읽어온다. 64 바이트 데이터의 앞부분부터 LXSDf T2 패킷 혹은 LXSDf UH1 패킷등 제품별로 지정된 LXSDf 규격으로 기록되어있으며, 이를 파싱하여 기기에서 전송한 구체적인 정보들을 추출한다.



ubpulse H3 의 호스트측 USB 수신버퍼 데이터 파싱.

앞의 설명처럼, ubpulse H3 가 연결된 호스트의 수신버퍼에는 LXSDf T2 TX 1 개의 패킷(총 19 바이트)이 배치되어 매번의 맥파 샘플링 시점 마다 USB 로 1 패킷이 탑재되어 호스트측으로 송신한다. 기기 내부적으로 맥파 샘플링은 초당 256 회 이뤄지고 있으므로 초당 256 회의 빠르기로 연속으로 USB 수신버퍼에 데이터가 기록되고 있다. LXSDf T2 TX 패킷에 탑재되어있는 데이터 내용은 아래 2 개의 규격서에 정의되어있으므로 아래 문서 참조하여 필요한 정보를 추출하여 활용한다.

문서아이디 : LXD13 . 제목 : **ubpulse H3 측정데이터 통신규격.**

다운로드 주소 : https://github.com/ubpulse/ubpulse-H3/raw/master/LXD13_ubpulseH3_CommunicationSpec_Measure.pdf

문서아이디 : LXD14 . 제목 : **ubpulse H3 HRV 검사 통신규격.**

다운로드 주소 : https://github.com/ubpulse/ubpulse-H3/raw/master/LXD14_ubpulseH3_CommunicationSpec_Analysis.pdf

LAXTHA USB 기기에서 USB HID 로레벨 버퍼로 데이터 전송하는 방식 이해.

LAXTHA USB HID 기기가 USB 호스트에서 인식완료되면 운영체제가 자동으로 65 바이트의 버퍼를 호스트측에 마련하게 된다. 기기에서 호스트로 데이터를 전송하면 본 65 바이트버퍼에 데이터가 도착되어있다. 이 버퍼이름을 호스트 입장에서 “수신버퍼”라고 한다.



LAXTHA USB HID 기기에서는 매번의 데이터 전송시 65 바이트 단위로 전송하는데, 첫번째 1 바이트는 더미 데이터이며 2 번째 바이트부터 LXSDf T2 혹은 LXSDf UH1 1 패킷을 기록하여 전송한다. 앱에서 읽을 때에도 1 번 데이터는 버리고 2 번째 데이터부터 데이터를 확보하여 사용해야한다. LAXTHA USB HID 기기는 기본 데이터블럭을 초당 256 번의 속도로 전송한다. 실시간 전송이 달성되어야 하는 시스템 이기 때문에 수신버퍼의 데이터를 앱에서 읽어가지 않는다면 덮어쓰기 하므로 앱에서 수신버퍼에서 데이터 읽기 처리하는 부분의 설계에 시간지연이 많이 생기지 않도록 주의를 기울여야한다.

호스트 측에서 USB HID 버퍼에서 데이터 읽는 방법.

호스트측 앱에서 장치 핸들을 확보하고 나면 USB 수신버퍼에 접근하여 데이터 값을 읽을 수 있어야 한다. 이는 VC++ 의 경우에는 아래 코드 샘플과 같이 ReadFile 을 이용한다. 이는 사용되는 언어별로 상이할 수 있으나 기본 개념은 동일하다.

호스트측에서 USB HID 버퍼에서 값을 읽어들이는 함수 예

```
// 지정한 디바이스 핸들에서 지정한 바이트 수 만큼 데이터를 ReadFile 하는 함수

short ReadFromDeviceHandle(HANDLE DeviceHandle, LPSTR lpByte, DWORD dwBytesToRead)
{
    BOOL    Result;
    DWORD   BytesRead;
    Result = ReadFile(DeviceHandle, // handle to device to write to
                     lpByte,       // point to data to write to file
                     dwBytesToRead, // number of bytes to read
                     &BytesRead,   // pointer to number of bytes read
                     NULL           // pointer to structure for data
                    );

    if(!Result)
    { // 읽기에 실패한 경우이다.
        // AfxMessageBox("에러");
    }
    return (Result);
}
```

호스트측에서 USB HID 수신버퍼에서 실시간 데이터 읽기 처리시 주의사항.

LAXTHA USB HID 기기는 실시간 데이터전송이 이뤄지는 시스템이므로 호스트 측에서 USB HID 수신 버퍼에서 데이터를 읽어오는 처리부에 주의가 필요하다.

호스트측 앱에서 USB HID 버퍼로부터 데이터 읽고 처리하는 부분은 별도의 스레드를 생성시켜서 앱의 주스레드와 분리되도록 하여, 실시간 데이터 수신에 방해받지 않도록 해준다.

USB HID 수신버퍼에서 한 번 데이터를 읽고처리하는데 소요되는 시간이 기기측에서 다음 데이터블럭을 전송하기 전에 완료해야한다. 다음 데이터 블럭이 전송되기 전에 완료되지 못하면 데이터 누락 및 프로그램이 오류상태에 빠질 수 있다. 이 문제를 그나마 부드럽게 처리할 수 있는 수단은 데이터 읽은 것을 잠시 저장해 둘 수 있는 Q-버퍼를 마련해 두는 것이다. 즉 USB HID 수신버퍼에서 읽은 데이터는 별다른 처리없이 Q-버퍼에 저장처리만 하도록 하며, 처리 스레드에서는 Q-버퍼에 있는 값을 읽어들여 후처리 하도록 한다. 즉, 통상적인 실시간 처리계에서와 같이 2 개의 스레드와 1 개의 Q-버퍼를 구비해 두는 것이 가장 안전한 방법이다.

스레드 1에서는 USB HID 수신버퍼에서 데이터를 읽어서 Q-버퍼에 기록하도록 하고, 스레드 2에서는 Q-버퍼에서 읽어서 후처리하도록 한다.

ubpulse H3 의 호스트 측 USB 송신버퍼.

ubpulse H3 가 호스트에 연결되면 65 바이트의 수신버퍼와 동시에 9 바이트의 송신버퍼도 자동으로 생성되며 호스트의 코드에서 송신버퍼에 데이터를 기록하면 ubpulse H3 로 해당 데이터가 송신되며 기기 제어 목적으로 활용된다.

9 바이트 중 첫바이트는 0 으로 기록하고, 이후 나머지 3 바이트에 LXSDf T2 RX 의 Cmd0, Cmd1, Cmd2 의 값을 기록하고, 송신버퍼의 나머지 5 개 바이트에는 모두 0 으로 기록한다. 송신버퍼에 기록된 데이터는 기기까지 자동으로 전송되며 해당 명령어에 따라 기기 내부적으로 처리 수행된다.

USB 송신버퍼	바이트 0	바이트 1	바이트 2	바이트 3	바이트 4	바이트 5	바이트 6	7	8
LXSDf T2 RX 배치.	0	Cmd0	Cmd1	Cmd2	0	0	0	0	0

명령전송의 예.

HRV 검사시작 명령 : Cmd0:128, Cmd1:1, Cmd2:0

HRV 검사중지 명령 : Cmd0:128, Cmd1:0, Cmd2:0

HRV 분석 데이터 요청 0 : Cmd0:128, Cmd1:2, Cmd2:0

등.

모든 명령별 Cmd0, 1, 2 의 값은 아래 문서 를 참조한다.

문서아이디 : LXD13 . 제목 : **ubpulse H3 측정데이터 통신규격.**

다운로드 주소 : https://github.com/ubpulse/ubpulse-H3/raw/master/LXD13_ubpulseH3_CommunicationSpec_Measure.pdf

문서아이디 : LXD14 . 제목 : **ubpulse H3 HRV 검사 통신규격.**

다운로드 주소 : https://github.com/ubpulse/ubpulse-H3/raw/master/LXD14_ubpulseH3_CommunicationSpec_Analysis.pdf

Revision History

Release Date	Doc. ID	Description of Change
2017-05-23	LXD146 V1	First release.