
neuroNicle API

Developer's Manual for neuroNicleAPI(v1.0)

Doc. ID. LXD92 V0

Release Date. 2015-05-11 .

Abstract – neuroNicle API 는 neuroNicle 장치와 윈도우 운영체제의 응용프로그램 사이에서 통신을 매개하는 DLL(Dynamic Link Library)이다. neuroNicle API 를 이용하여, neuroNicle 장치로부터 전송되는 실시간 측정 데이터를 응용프로그램에서 활용가능하다. API 내부적으로 장치제어, 실시간 처리 스레드 등이 구현되어 있어, 쉽고 빠른 응용프로그램 개발이 가능하다..

목차

NEURONICLE API 개요.	4
NEURONICLE API 주요특징.	4
NEURONICLE API 구성 파일.	5
응용프로그램에서 NEURONICLE API 임포트 방법.	6
NEURONICLE API 파일들 개발프로젝트 폴더에 복사하기.	6
응용프로그램의 실행파일이 있는 폴더에 3 개 API 파일 복사.	6
응용프로그램의 소스가 있는 폴더에 2 개의 API 파일 복사.	7
응용프로젝트(예: vc++) 소스에 임포트 코딩 코드 추가.	8
스트림데이터 구조체.	9
스트림 구조체 멤버의 파형 및 이벤트 스트림 데이터.	9
스트림 데이터 1 차원배열 인덱스 접근법.	10
스트림 데이터 표현예.	10
API 에서 발생하는 메시지.	11
API 에서 전송되는 메시지 인자 wPARAM 에 기록된 데이터의 공통규격.	11
메시지타입 ID 0	12
메시지타입 ID 1	13
함수 호출순서	15
전체 함수호출 코드 예.	16
FUNCTION REFERENCES.	17
OPENAPI	17
설명	17
인자	17
반환값	17
OPENDEVICE	18
설명	18
인자	18
반환값	18

SETMESSAGEDEVICE	19
설명	19
인자	19
반환값	19
STARTSTREAM	20
설명	20
인자	20
반환값	20
GETSTREAMDATA	21
설명	21
인자	21
반환값	21
EVENTMARKINGONSTREAM	22
설명	22
인자	22
반환값	22
STOPTSTREAM	23
설명	23
인자	23
반환값	23
CLOSEDEVICE	24
설명	24
인자	24
반환값	24
CLOSEAPI	25
REVISION HISTORY.....	26

neuroNicle API 개요.

neuroNicle 장치가 PC(윈도우 운영체제) 와 USB, Bluetooth 인터페이스 등으로 연결되어있는 상태에서, PC 응용프로그램에서 neuroNicle 장치와 통신부분을 쉽게 구현할 수 있는 수단이 neuroNicle API 이다. neuroNicle API 에서 neuroNicle 브랜드의 모든 모델의 제품들과 통신가능하다. neuroNicle API 는 Microsoft Visual C++ 로 제작된 DLL 형식이다.

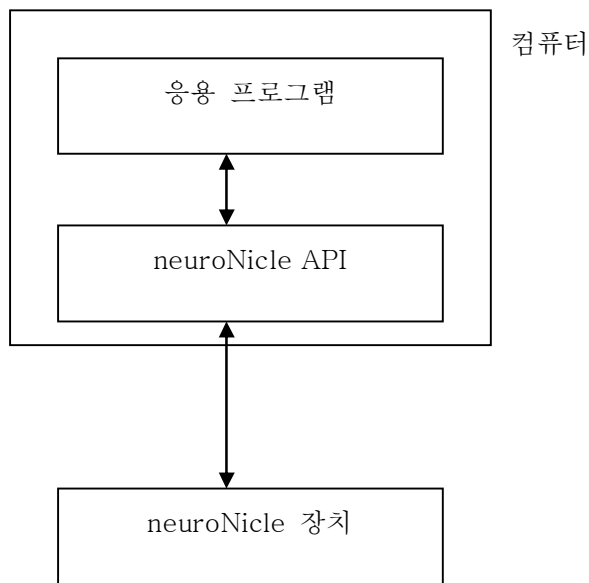


그림 1. 장치-neuroNicle API –응용프로그램과의 관계

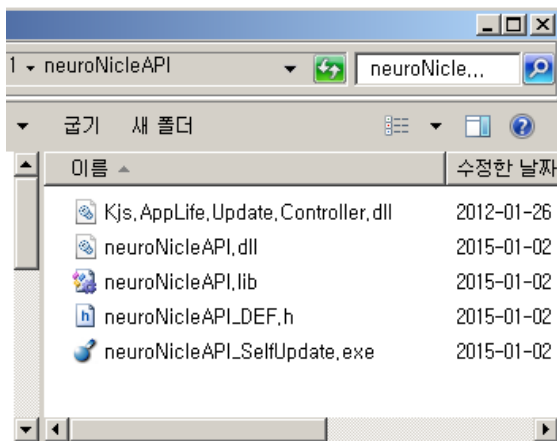
neuroNicle API 주요특징.

- 모든 neuroNicle 제품군들과 통신가능. (neuroNicle E1, neuroNicle E2, neuroNicle B102 etc.)
- Microsoft Visual C++ 2010 으로 제작된 MFC Regular DLL.
- Self Updating API (SUA)
- -- API 수정 사항 발생시 네트워크를 통한 자동 업데이트가 이뤄지므로 유지보수 편리.
- -- 응용프로그램에서 SUA 기능 활성 비활성 선택가능하여 응용프로그램이 네트워크 기반 자동 업데이트 기능이 구현되어 있는 경우엔 비활성처리 가능.

neuroNicle API 구성 파일.

배포파일명 : neuroNicleAPI.zip

다운로드 받은 파일 neuroNicleAPI.zip 을 압축해제 하면 아래 표와 같은 파일들을 볼 수 있다.



파일명	설명
neuroNicleAPI.DLL	응용프로그램의 실행파일경로에 복사해둔다.
neuroNicleAPI.LIB	응용프로그램에서 implicit linking 시켜서 사용한다.
neuroNicleAPI_DEF.h	neuroNicleAPI.DLL 에서 제공하는 함수 선언, 변수정의가 있다. DLL 에서 제공하는 함수를 사용하기 위하여 응용프로그램에서 인클루드 시켜서 사용한다.
neuroNicleAPI_SelfUpdate.exe	API Self Update 기능 관련 파일 2 개.
Kjs.AppLife.Update.Controller.dll	응용프로그램 실행시 네트워크 자동업데이트 기능으로 neuroNicleAPI.DLL 의 최신버전이 있는지 점검하고 자동갱신 처리기능.

응용프로그램에서 neuroNicle API 임포트 방법.

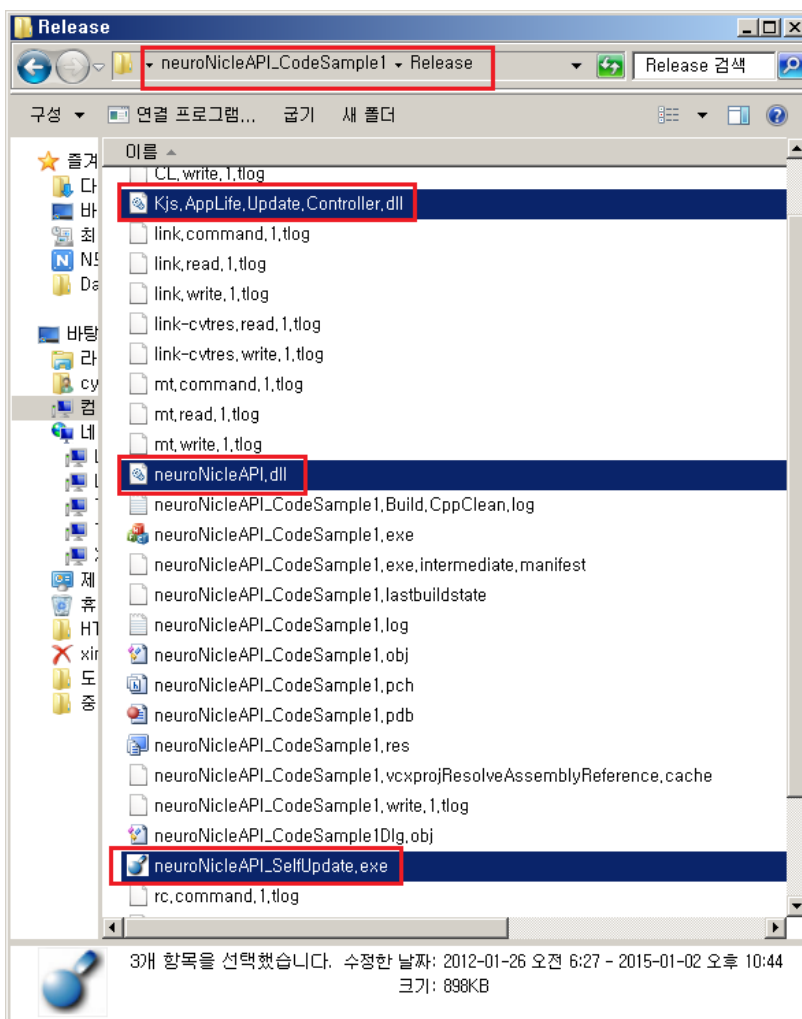
neuroNicle API 파일들 개발프로젝트 폴더에 복사하기.

응용프로그램의 실행파일이 있는 폴더에 3 개 API 파일 복사.

응용프로그램의 실행파일이 있는 경로에 아래 3 개의 파일을 복사한다.

- neuroNicleAPI .DLL
- neuroNicleAPI_SelfUpdate.exe
- Kjs.AppLife.Update.Controller.dll

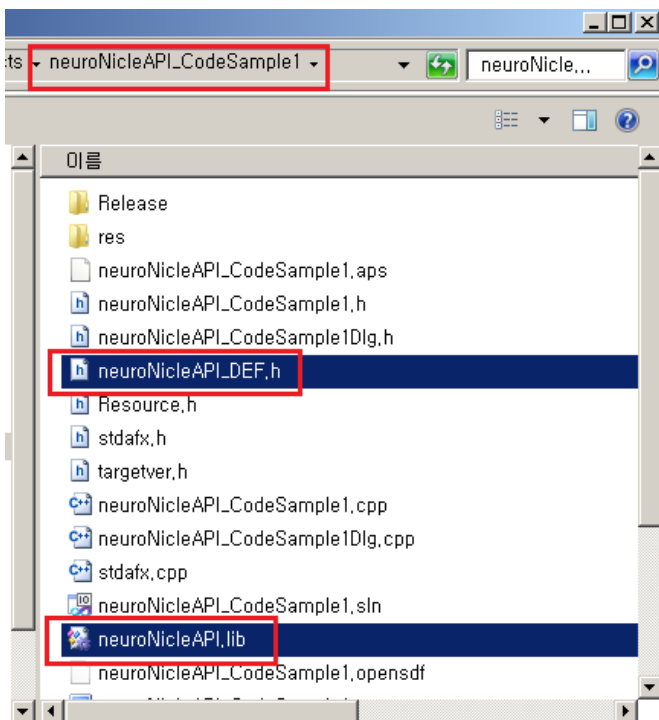
아래 그림은 위 3 개 파일이 응용프로그램의 release 폴더에 복사된 모습을 보이고 있다. 위 3 개 파일은 응용프로그램 개발이 완료되어 최종 사용자에게 배포될때도 필수 포함되어 같이 배포되어야 한다.



응용프로그램의 소스가 있는 폴더에 2 개의 API 파일 복사.

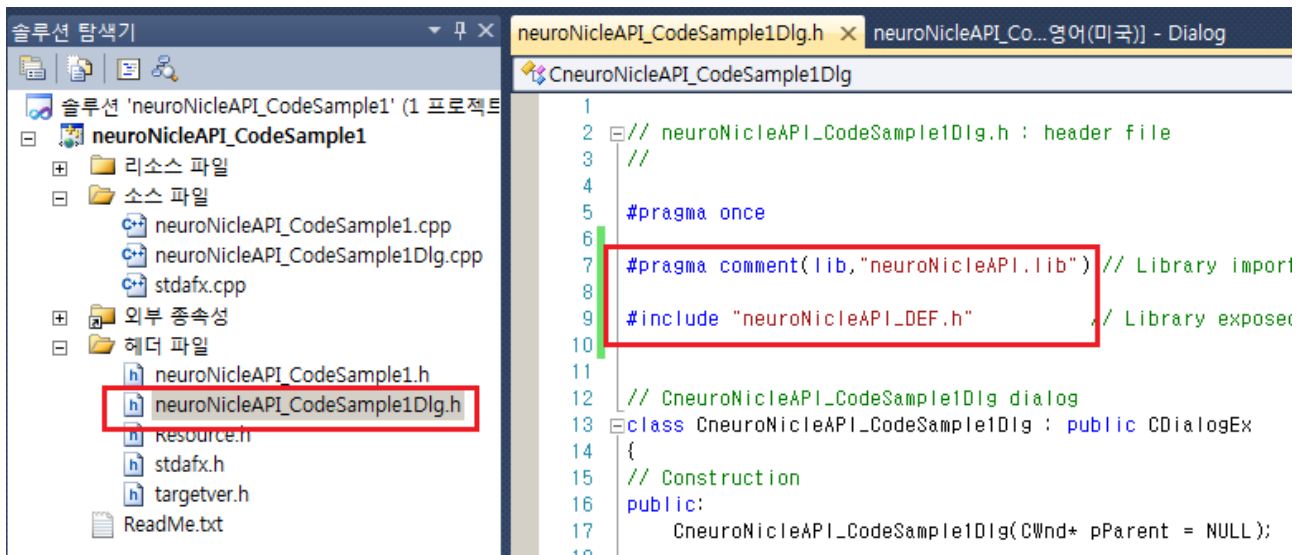
아래 2 개 파일은 응용프로그램의 소스가 있는 폴더에 복사한다.

- neuroNicleAPI.LIB
- neuroNicleAPI_DEF.h



응용프로젝트(예 : vc++) 소스에 임포팅 코드 추가.

아래 그림의 예에서 처럼 API 의 헤더파일을 인클루드 처리하고, 라이브러리 파일인 neuroNicleAPI.lib 를 링크처리시킨다.



이로써, neuroNicleAPI 를 응용프로그램 개발에서 활용하기 위한 임포팅 처리는 완료되었고, 이제 API 에서 제공하는 기능을 사용가능하다.

스트림데이터 구조체.

API에서 스트림데이터를 정의하는 데이터 형식은 구조체 `stStreamData_nnAPI` 이다. 구조체 `stStreamData_nnAPI` 의 타입정의는 API와 같이 제공되는 `neuroNicleAPI_DEF.h` 파일에서 제공되고 있다.

응용프로그램에서 구조체 변수를 선언하고 이 변수를 통하여 API 에서 제공하는 실시간 데이터를 확보하여 활용하게 된다.

사용예.

```
stStreamData_nnAPI stStreamData; // 구조체 변수선언.
```

```
OpenDevice_neuroNicleAPI( , &stStreamData, ); // OpenDevice 함수 호출시 구조체변수 주소전달.
```

```
GetStreamData_neuroNicleAPI( , &stStreamData); // 응용프로그램의 메시지 핸들러 내부에서 구조체변수로 스트림데이터를 모두 확보.
```

스트림 구조체 멤버의 파형 및 이벤트 스트림 데이터

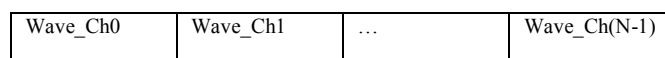
스트림 구조체 멤버중 가장 중요한 변수는 1 차원 배열인 `float* Wave_StreamData_CS`, `unsigned int* Event_StreamData_CS` 2 개의 변수이다. 장치에서 측정된 파형 데이터는 `Wave_StreamData_CS[]` 에 기록되며, 파형외의 이벤트 데이터는 `Event_StreamData_CS[]` 에 저장된다.

1 차원 배열 `Wave_StreamData_CS` 에 파형 데이터가 배치된 모습과 `Event_StreamData_CS` 에 이벤트 데이터가 배치된 모습은 아래 그림과 같다.

`Wave_StreamData_CS`

`N = Wave_NumChannel`

자료형: `double`

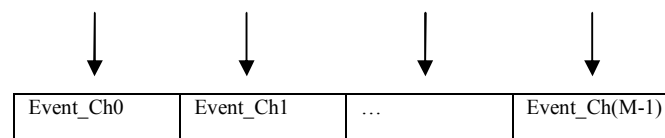


모든 Wave 채널, Event 채널에는 동일한 수량(`NumSampleReturn`)의 샘플링데이터가 기록되어있음.

`Event_StreamData_CS`

`M=Event_NumChannel`

자료형 : `unsigned int`



스트림 데이터 1 차원배열 인덱스 접근법.

스트림 구조체 멤버 변수인 Wave_StreamData_CS , Event_StreamData_CS 는 모두 1차원 배열이며, 모든 채널에 대한 샘플링 데이터가 기록되어있다. 채널인덱스와 샘플인덱스로 해당값에 접근하기 위한 배열인덱싱 방법.

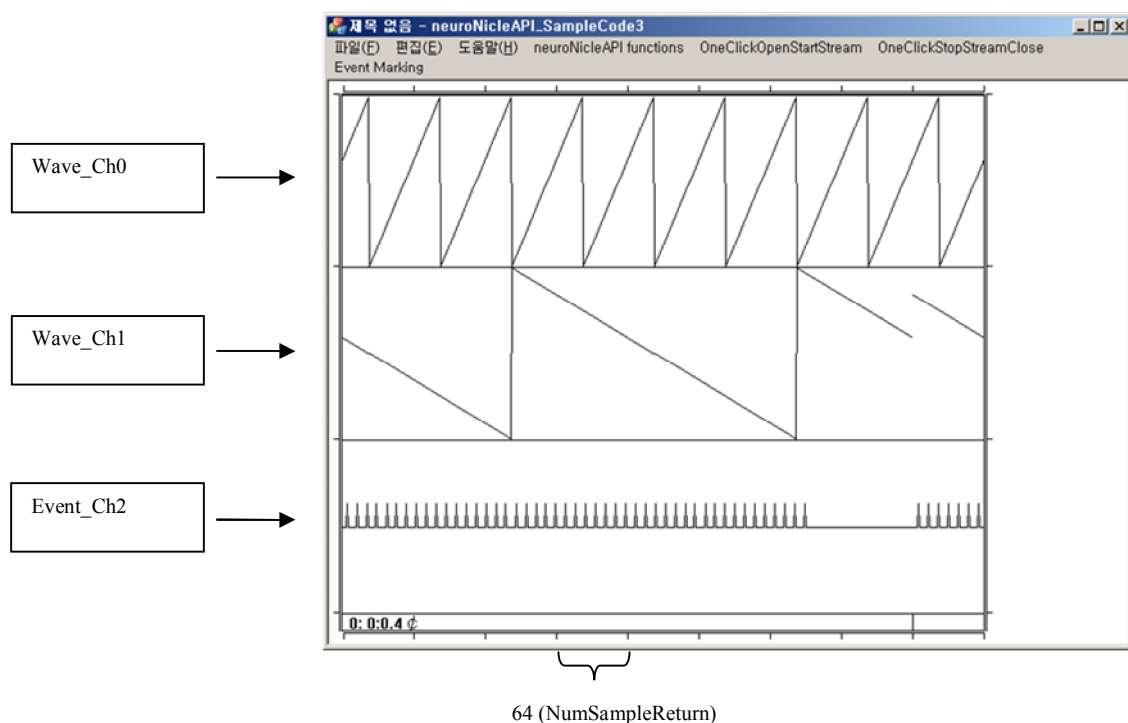
Wave_StreamData_CS [wave_ch_idx * NumSampleReturn + sample_idx]

Event_StreamData_CS [event_ch_idx * NumSampleReturn + sample_idx]

NumSampleReturn : 구조체 stStreamData_nnAPI 의 멤버 변수 NumSampleReturn 에서 받을 수 있음.

스트림 데이터 표현예.

아래 응용프로그램의 출력은 NumSampleReturn 이 64 이고, Wave 채널이 2 이면서, Event 채널 1 개 를 표현한 예이다.



API 에서 발생하는 메시지.

API 에서 전송되는 메시지 인자 wParam 에 기록된 데이터의 공통규격.

API 에서 메시지 전송시 wParam 에서는 unsigned int(4 바이트) 형의 정수를 전송하며, wParam 의 하위 바이트부터 wParam_Byte0, wParam_Byte1, wParam_Byte2, wParam_Byte3 이라고 하였을때, 아래표와 같은 데이터가 기록되어 전송된다.

wParam 의 바이트	기록된 데이터.	설명.
wParam_Byte3 wParam_Byte2 (wParam 의 상위 2Bytes)	장치핸들링 ID 값의 범위 1~65535	.API 를 이용하여 동시에 2 개 이상의 장치와 통신하는 경우, API 에서 전송된 메시지를 수신한 응용프로그램에서 wParam 으로 전송된 장치핸들링 아이디값을 확인하여 어떤 장치에서 전송된 메시지인지 판단할 때 사용된다. 장치핸들링 ID 는 OpenDevice 함수 호출시 API 에서 자동생성되어 반환되는 정수 값이며, 통신개설된 장치에 할당되는 고유한 값이다.
wParam_Byte1	메시지타입 ID. 값의 범위 0~255	API 에서 전송된 메시지가 어떤 “메시지타입 ID”에서 발생한 메시지인지 식별이 필요한 경우 활용된다. 관련 : SetMessageDevice 함수호출시 메시지타입 ID 별로 수신받을 윈도우, 메시지아이디, 메시지 수신여부를 지정하게 되어있다.
wParam_Byte0	메시지타입 서브 ID. 값의 범위 0~255	1 개의 메시지타입 ID 에서도 여러 종류의 데이터가 메시지로 전송되며, 메시지타입서브 ID 를 확인하여 전송된 메시지가 어떤 종류의 데이터인지 식별할 때 필수로 활용된다.

메시지타입 ID 0 .

메시지타입 ID 0 은 “스트림데이터” 실시간 처리와 관련된 정보 전송 목적으로 사용된다.

메시지타입 ID 0 의 메시지 1 번 인자 wParam 의 상위 3 바이트는 아래표와 같이 동일한 값이 기록되었었으며, wParam 의 최하위 바이트인 “메시지타입 sub-ID”와 lParam 에 기록된 값이 상황마다 다르게 기록되어 메시지 전송된다.

wParam Byte3, 2 (장치핸들링 ID)	wParam_Byte1 (메시지타입 ID)
OpenDevice 반환값	0

wParam Byte 0 (메시지타입 sub-ID)	lParam	의미
0	0	API 측에 요청한 반환샘플링 수량만큼 데이터가 모두 확보된 시점에 발생. 응용프로그램측에서 메시지 핸들러 내에서 GetStreamData 함수를 호출하여 구조체변수(stStreamData_nnAPI 타입) 로 스트림 데이터를 확보처리하고 데이터를 활용한다. 다음 메시지 수신전까지 데이터 처리를 모두 완료해야함.
	Reserved	-
1	Reserved	-
	Reserved	-
2	Reserved	-
	Reserved	-
3	Reserved	-
	Reserved	-

메시지타입 ID 1 .

메시지타입 ID 1의 메시지에는 실시간성이 크게 중요치 않은 정보들이 메시지로 전송된다.

메시지타입 ID 1의 메시지 인자 wParam의 상위 3 바이트는 아래표와 같이 동일한 값이 기록되어 있으며, wParam의 최하위 바이트인 “메시지타입 sub-ID”와 lParam에 기록된 값이 상황마다 다르게 기록되어 메시지 전송된다.

wParam Byte3, 2 (장치핸들링 ID)	wParam_Byte1 (메시지타입 ID)
OpenDevice 반환값	1

wParam Byte 0 (메시지타입 sub-ID)	lParam	의미
0	0	측정 불가능 상태.(장치의 센서가 인체에 착용되지 못함)
	1	측정 가능 상태.(장치의 센서가 인체에 착용됨)
1	0	USB 연결끊김. 아래 상세설명 필수 참조.
	Reserved	-
2	Reserved	-
	Reserved	-
3	Reserved	-
	Reserved	-

상세설명 : USB 연결끊김.

OpenDevice 함수가 실행되어 장치와 통신 가능 상태에서, USB 연결이 끊긴 경우 발생하는 메시지이며

메시지타입 sub-id = 1, lParam = 0 으로 전달된다.

응용 프로그램에서 조치해야할 사항

응용프로그램의 메시지타입아이디 1 수신처리부에서 “USB 연결끊김”을 수신한 경우 반드시 **CloseDevice** 함수를 호출해야한다. 사용자가 USB 연결을 정상화 한 이후 OpenDevice -> SetMessageDevice -> StartStream의 순서로 함수 호출하여 정상사용가능하다.

코드예. - 응용프로그램의 USB 연결 끊김 메시지 수신처리부

```
메시타입아이디1핸들러(WPARAM wParam, LPARAM lParam)
{
    unsigned char messagetype_sub_id = (unsigned int)wParam; // wParam 의 최하위 1바이트값(MessageType Sub-ID) 을 받았다.

    switch(messagetype_sub_id)
    {

        case 1: // 장치오류 정보.

            if((unsigned int)lParam == 0) // OpenDevice 이후 USB 연결해제됨. USB연결후 neuroNicle API
            OpenProcess 시행해야함. 프로그램 종료하지 않아도 됨.
            {

                CloseDevice_neuroNicleAPI(DeviceHandlingID); // 장치 닫기.- 필수 수행해야함.

                CloseApi_neuroNicleAPI(); // neuroNicle API닫기.

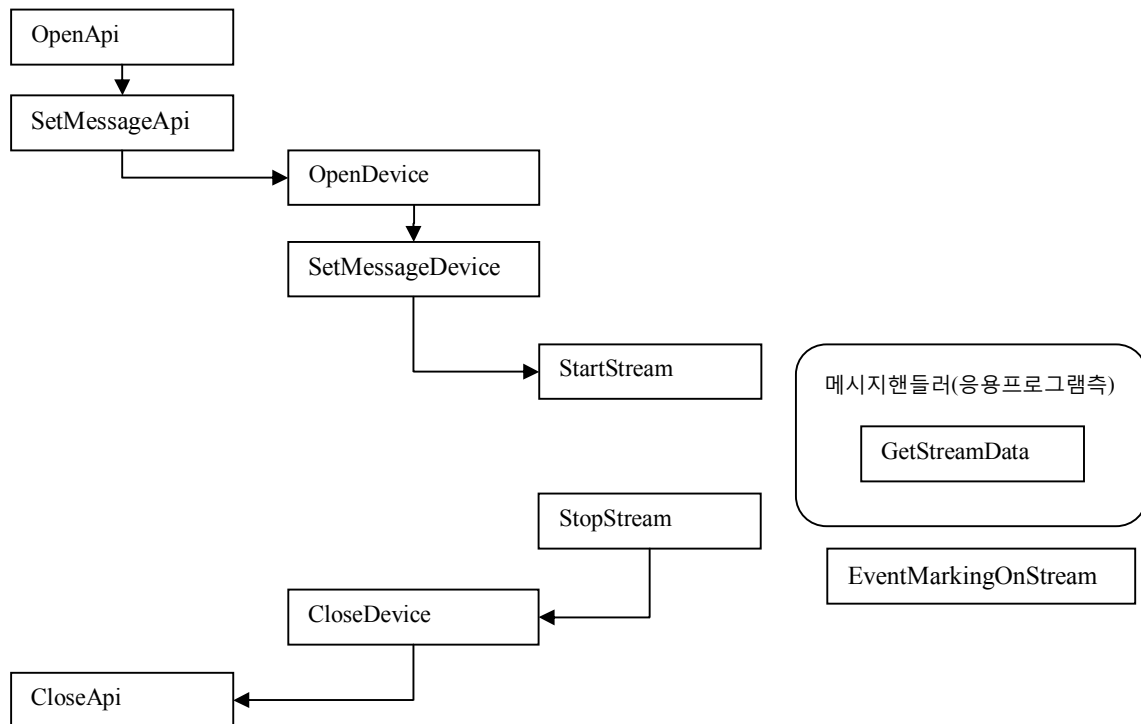
            }

            break;

    }
}
```

USB 연결끊김 메시지 수신시 CloseDevice 함수는 반드시 호출해야하며, 이후 CloseApi 함수는 필수요구사항은 아니다. 만일 CloseDevice 와 CloseApi 모두 호출했다면 , 사용자가 USB 연결 정상화 조치한 이후 정상 측정을 위하여 응용프로그램에서는 OpenApi -> OpenDevice -> SetMessageDevice -> StartStream 의 순서로 OpenApi 부터 시작하여 API 함수 호출하면된다.

함수 호출순서



GetStreamData 함수 호출지점.

StartStream 호출이후에 API에서는 메시지타입 0으로 지정된(함수 SetMessageDevice로 지정) 메시지가 전송되며 응용 프로그램에서는 메시지타입 0을 처리하는 메시지 핸들러가 구현되어야 한다. 메시지 핸들러 내에서 매번의 메시지 수신될때마다 GetStreamData 함수를 호출하여 스트림데이터를 확보하여 응용프로그램에서 스트림 데이터를 활용한다.

OpenApi 호출 이후 SetMessageAPI는 호출하지 않고 OpenDevice를 호출해도 되며, 나머지 함수들은 위 그림의 순서대로 모두 호출해야 스트림데이터를 정상수신할 수 있는 상태가 됨.

전체 함수호출 코드 예.

아래 Open 함수내에서는 OpenApi -> OpenDevice -> SetMessageDevice -> StartStream 까지 일괄호출되는 경우를 보이고 있으며, Close 함수내에서는 StopStream -> CloseDevice -> CloseApi 까지 일괄호출되는 예를 보이고 있다.

```
void Open ()
{
    OpenApi_neuroNicleAPI(1,0);

    NumSampleReturn = 4;

    int retv = OpenDevice_neuroNicleAPI(256,&stStreamData,NumSampleReturn);
    if(retv > 0)
    {
        DeviceHandlingID = retv;

        SetMessageDevice_neuroNicleAPI(DeviceHandlingID,0,this->m_hWnd,WM_nnAPI_MSGTYPEID_0,1);

        StartStream_neuroNicleAPI(DeviceHandlingID);
    }
}

void Close()
{
    if(DeviceHandlingID > 0)
    {
        StopStream_neuroNicleAPI(DeviceHandlingID);

        CloseDevice_neuroNicleAPI(DeviceHandlingID);
    }
    CloseApi_neuroNicleAPI();
}
```


Function References

OpenApi

`int` OpenApi_neuroNicleAPI(`int` api_window, `int` api_selfupdate, `int` mode)

설명

neuroNicle API 를 사용하기 위해서 가장 먼저 호출되어야 한다. OpenApo 호출이후에 API 에서 제공되는 다른 함수 호출이 가능하다.

인자

인자	가능한 값.	설명
<code>int</code> api_window	0 : API 윈도우 비활성화. 1 : API 윈도우를 활성화. (기본값 : 0)	OpenApi 호출시 API 윈도우를 보이기 여부 지정. 앱 개발중에는 1 로 설정하여 API 윈도우가 보이는 상태로 하고, 앱 개발완료된 경우 API 윈도우를 보이지 않는 상태로 한다.
<code>int</code> api_selfupdate	0: check update 실행않음. 1: check update 실행함. (기본값 : 1)	neuroNicle API 는 네트워크를 통한 자동원격업데이트가 지원되어 항상 최신버전의 API 파일로 유지할 수 있다. 이 값을 1 로 하여 OpenApi 호출하면 Check for Update 창이 실행된다. 개발 중의 앱이 원격업데이트 기능이 없는 경우엔 1 로 설정한다. neuroNicle API 의 신규업데이트를 앱에서 직접 관리하려는 경우에는 0 으로 설정한다.
<code>int</code> mode	(기본값 : 0)	Don't care

반환값

반환값	의미	설명
1	성공	
-1	실패	
-2	실패. 중복호출	OpenApi 가 이미 한 번 호출되어 정상수행된 이후에는 재 호출 불가. OpenApi 를 다시 실행하려면 CloseApi 호출 이후에 가능.

OpenDevice

`int` OpenDevice_neuroNicleAPI(`int` LXDeviceID, `stStreamData_nnAPI*` p_stream_data, `int` numsample_return, `int` mode)

설명

1 개의 장치와 통신 가능 상태 달성.

함수의 인자로 전달된 LXDeviceID 에 해당하는 장치와 통신가능 상태를 달성하고, 장치에서 실시간 전송되는 데이터를 앱측에서 수신하기 위한 stStreamData_nnAPI 로 정의되어 있는 구조체 변수를 정의하고, 구조체 멤버 Stream_NumSample 에 앱에서 수신할 샘플링 수량을 지정한 후, 구조체 변수 포인터를 함수인자로 전달해야한다. 본 함수가 성공적으로 실행되면 장치와 통신가능 상태가 달성되며, 함수의 반환값으로 DeviceHandlingID(장치핸들링 아이디) 가 반환된다. 장치핸들링 아이디는 이후 장치제어함수들 호출시 입력 인자로 필수 요구되므로 앱에서 확보 저장해둬야 한다.

인자

인자	가능한 값.	설명
<code>int</code> LXDeviceID	1~ 65535 범위의 장치 식별 고유아이디.	통신 개설할 장치를 지정하는 것이며, 장치 모델별 고유 아이디는 XXX 에서 확인할 수 있다.
<code>stStreamData_nnAPI*</code> p_stream_data	-	API 의 스트림 전송 상태에서 “스트림데이터”를 받기 위한 구조체 변수를 선언하고 변수의 주소를 함수호출시 전달한다. 예. stStreamData_nnAPI stStreamData; // 구조체 변수 선언. OpenDevice_neuroNicleAPI(, &stStreamData, ,); // 구조체변수 주소전달.
<code>int</code> numsample_return	1 이상	앱에서 “스트림데이터” 1 회 수신시 샘플(각채널의 데이터 수량)수량 지정. 만일 1 을 기록했다면, API 에서는 장치로부터 각 채널당 1 샘플링의 데이터가 수집될 때마다 앱측으로 스트림데이터가 전송된다.
<code>int</code> mode	(기본값 : 0)	Don't care

반환값

반환값	의미	설명
1 이상	성공	OpenDevice 호출결과 성공적인 경우, Device Handling ID(값의 범위 1~65535 사이의 임의 정수)가 반환되며, 앱에서 필수로 저장해두고 있어야 한다. 장치 제어함수 호출시 인자로 필수 요구되는 값이다.
-1	지원장치 아님.	LXDeviceID 에 해당하는 장치가 neuroNicle API 에서 지원하는 장치 아님.
-2	중복호출오류	동일장치가 OpenDevice 로 이미 통신개설된 상태에서 OpenDevice 가 호출된 경우.
-3	장치없음	LXDeviceID 에 해당하는 장치가 발견되지 않음.PC 에 연결되어있지 않음.
-10	함수호출순서 오류	OpenApi 호출이후에 OpenDevice 호출되어야 함.
-20	인자값범위오류	numsample_return (스트림 반환 수량) 값의 범위 1~64 까지만 가능함.

SendMessageDevice

```
int SendMessageDevice_neuroNicleAPI(int device_handling_id,int msgtype_id,HWND hwnd_msgrcv,int msg_id,int onoff);
```

설명

OpenDevice 로 통신개설된 장치에서 전송되는 메시지 설정한다.

인자

인자	가능한 값	설명
int device_handling_id	OpenDevice 함수 반환값.	OpenDevice 가 성공적으로 수행된 경우 반환된 값을 인자로 전달한다.
int msgtype_id	0~9	메시지타입아이디
HWND hwnd_msgrcv	윈도우 핸들	API 에서 전송되는 메시지(msgtype_id 에 해당하는)를 수신할 응용프로그램측의 윈도우 핸들을 전달한다.
int msg_id	메시지	neuroNicleAPI_DEF.h 에 메시지타입아이디별로 정의되어있는 메시지 아이디를 전달한다. 예. 메시지타입아이디 0 인 경우, SendMessageDevice_neuroNicleAPI(DeviceHandlingID,0,this->m_hWnd,WM_nnAPI_MSGTYPEID_0,1);
int onoff	1: 메시지 전송 켜기 0: 메시지 전송 끄기	msgtype_id 에 해당하는 메시지 발생유무.

반환값

반환값	의미	설명
1	성공	메시지 설정성공.
-3	장치없음	device_handling_id 로 연결된 장치없음.
-10	함수호출순서 오류	OpenApi 호출이후에 OpenDevice 호출되어야 함.

StartStream

`int StartStream_neuroNicleAPI(int device_handling_id)`

설명

OpenDevice 로 통신개설된 장치에서 스트림데이터 전송시작시킨다.

본 함수 성공적으로 호출되면, API 에서 메시지타입아이디 o 에 해당하는 메시지를 전송하게 되며 응용프로그램측에서는 메시지 핸들러에서 스트림데이터 수신처리를 해야한다.

인자

인자	가능한 값	설명
<code>int device_handling_id</code>	OpenDevice 함수 반환값.	OpenDevice 가 성공적으로 수행된 경우 반환된 값을 인자로 전달한다.

반환값

반환값	의미	설명
1	성공	메시지 설정성공.
-3	장치없음	device_handling_id 로 연결된 장치없음.
-10	함수호출순서 오류	OpenApi 호출이후에 OpenDevice 호출되어야 함.

GetStreamData

```
int GetStreamData_neuroNicleAPI(unsigned int messge_wparam, stStreamData_nnAPI* p_streaminfo);
```

설명

응용프로그램 측의 스트림수신 메시지 핸들러내에서 호출되어 스트림데이터를 확보한다.

인자

인자	가능한 값.	설명
unsigned int messge_wparam	메시지 수신 인자 wParam.	스트림 수신 메시지 (메시지타입아이디 0) 로 전달된 인자중 wParam 을 unsigned int 로 형변환하여 입력한다.
stStreamData_nnAPI* p_streaminfo	-	OpenDevice 시 인자로 전달한 동일한 구조체 변수의 주소 입력. 예. GetStreamData_neuroNicleAPI(, &stStreamData);

반환값

반환값	의미	설명
1	성공	메시지 설정성공.
-3	장치없음	device_handling_id 로 연결된 장치없음.
-10	함수호출순서 오류	OpenApi 호출이후에 OpenDevice 호출되어야 함.

EventMarkingOnStream

`int` EventMarkingOnStream_neuroNicleAPI(`int` device_handling_id, `unsigned int` event_value);

설명

API에서 스트림전송중에 응용프로그램 측의 특정 이벤트(키보드 누름, 화면제시시점등) 정보를 event_value로 입력하면 수신된 이벤트 스트림 데이터 구조체의 멤버 Event_StreamData_CS 의 채널3에 기록되어 같이 전송된다.

인자

인자	가능한 값.	설명
<code>int</code> device_handling_id	OpenDevice 함수 반환값.	OpenDevice 가 성공적으로 수행된 경우 반환된 값을 인자로 전달한다.
<code>unsigned int</code> event_value	0~65535	응용프로그램에서 여러 이벤트를 식별할 수 있도록 값을 정의해서 입력함. 예. 키보드 왼쪽화살표 키 누름 : 2 키보드의 오른쪽 화살표 키 누름 : 3

반환값

반환값	의미	설명
1	성공	메시지 설정성공.
-3	장치없음	device_handling_id 로 연결된 장치없음.
-10	함수호출순서 오류	OpenApi 호출이후에 OpenDevice 호출되어야 함.

StoptStream

```
int StopStream_neuroNicleAPI(int device_handling_id);
```

설명

API에서의 스트림전송을 중지한다. 스트림 전송을 다시 시작하려면 StartStream 호출한다.

인자

인자	가능한 값.	설명
int device_handling_id	OpenDevice 함수 반환값.	OpenDevice가 성공적으로 수행된 경우 반환된 값을 인자로 전달한다.

반환값

반환값	의미	설명
1	성공	메시지 설정성공.
-3	장치없음	device_handling_id로 연결된 장치없음.
-10	함수호출순서 오류	OpenApi 호출이후에 OpenDevice 호출되어야 함.

CloseDevice

```
int CloseDevice_neuroNicleAPI(int device_handling_id);
```

설명

OpenDevice 로 통신개설했던 장치연결을 해제한다.

인자

인자	가능한 값.	설명
int device_handling_id	OpenDevice 함수 반환값.	OpenDevice 가 성공적으로 수행된 경우 반환된 값을 인자로 전달한다.

반환값

반환값	의미	설명
1	성공	메시지 설정성공.
-3	장치없음	device_handling_id 로 연결된 장치없음.
-10	함수호출순서 오류	OpenApi 호출이후에 OpenDevice 호출되어야 함.

CloseApi

`int CloseApi_neuroNicleAPI()`

설명

neuroNicle API 를 더 이상 사용하지 않는 경우 가장 마지막으로 호출한다.

반환값

반환값	의미	설명
1	성공	
-10	실패	OpenApi 호출이후에 CloseApi 호출가능함.

Revision History

Release Date	Doc. ID	Description of Change
2015-x-	LXD92 V1.0	First release.