
LXSMWD2 개발자용 설명서

LXSMWD2 : USB HID 기반 맥파장치 통신용 DLL

Doc. ID. LXD21 V1

Release Date. 2012-10-22 .

Abstract – 본 설명서는 DLL (명칭 : LXSMWD2) 를 활용하기 위한 방법에 대해서 다룬다. LXSMWD2 는 USB HID (human interface device)기반의 맥파장치와 통신이 가능한 Win32 API 형식의 DLL 이다.

목차

LXSMWD2 개요.....	3
LXSMWD2 로 통신가능한 장치.	3
LXSMWD2 구조.....	4
LXSMWD2 구성요소.....	4
LXSMWD2 활용하기 위한 핵심정보 가이드.....	5
LXSMWD2 의 핵심작동방식.	6
스트림상태에서 DLL 이 발생하는 메시지.	8
메시지타입인덱스 2 : 맥파센서의 연결상태정보.....	9
메시지타입인덱스 3 : 맥파센서에 손가락 배치 상태정보	10
메시지타입인덱스 5 : 스트림실행 중 장치연결 해제	10
스트림데이터의 구조	11
채널을 부분적으로 선택한 경우의 배열의 데이터구조.	11
스트림데이터 관련 코딩 상세설명.....	13
메시지로 전달된 인자로부터 float 형 배열로 데이터를 받는 방법에 대한 설명.....	14
함수설명서.....	16
함수호출순서.....	18
LXSMWD2 활용 예제 VC++ 6.0 프로젝트	19
TEST_LXSMWD2 사용법.	19
심박수와 심박 시간격을 메시지로 수신하는 기능시험.	21

LXSMWD2 개요.

LXSMWD2 는 USB HID(Human Interface Device) 기반의 맥파 측정장치들로부터 실시간 데이터수집을 담당하는 WIN32 API 형식의 DLL 이다.

USB HID 기반의 맥파 측정이 가능한 제품은 ubpulse T1, RP920 등이 있다. 이들 제품들은 공통적으로 USB 로 컴퓨터와 연결되어 작동이 가능하다. 컴퓨터 측에서 개발하는 응용프로그램 에서 USB 통신처리와 같은 하드웨어적인 지식이 요구되지 않고, S/W 작업만으로 장치제어 및 데이터 수집이 가능하다.

LXSMWD2 로 통신가능한 장치.

ubpulse T1



RP920



LXSMWD2 구조.

모프로그래밍과 LXSMWD2 와 제품간의 관계는 아래 그림과 같다. 장치와의 모든 통신은 LXSMWD2 를 경유한다.

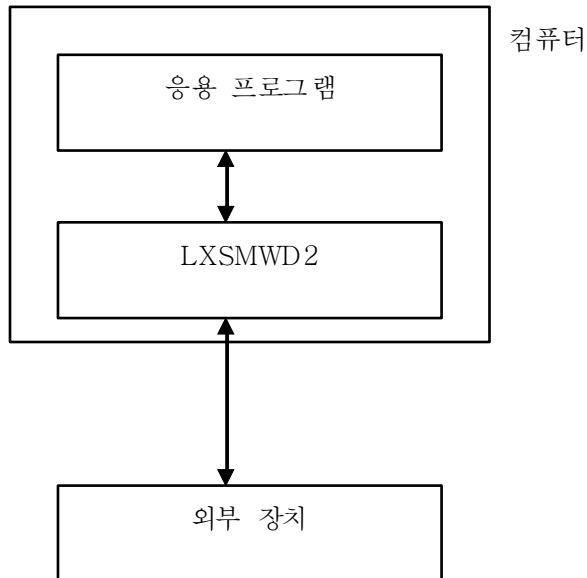


그림 1. 장치-DL-응용프로그램 관계

LXSMWD2 구성요소

LXSMWD2 는 아래 표와 같이 3 개의 파일로 구성되어있다.

파일명	설명
LXSMWD2.DLL	응용프로그램의 실행파일경로에 복사해둔다.
LXSMWD2.LIB	응용프로그램에서 implicit linking 시켜서 사용한다.
LXSMWD2.h	본 DLL 에서 제공하는 함수를 사용하기 위하여 응용프로그램에서 인클루드 시켜서 사용한다.

추가배포되는 Test_LXSMWD2 라는 VC++6.0 프로젝트 파일이 있으며, 본 DLL 을 사용하는 예제를 보이고 있음.

LXSMWD2 활용하기 위한 핵심정보 가이드

LXSMWD2 를 활용하기 위해서는 DLL 이 제공하는 함수와 스트림전송모드에서 DLL 에서 발생하는 메시지에 관련한 정보만 명확히 알면 응용프로그램의 코딩이 가능하다.

알아야할 정보.	설명	상세정보
LXSMWD2 의 핵심작동방식	DLL 을 응용프로그램에서 전체적으로 어떤방식으로 사용해야 하는가?	페이지 6,7
메시지	스트림 모드에서 전송되어 오는 메시지의 의미 및 전송되는 값의 의미.	페이지 8 ~ 10
스트림데이터를 저장하고있는 배열의 데이터 저장방식	스트림작동중 전송되는 메시지 타입인덱스 o 의 인수로 전달되는 것은 스트림데이터를 저장하고 있는 float 형 배열의 시작 번지이다. 본 배열에 데이터가 어떤 구조로 배치되어있는가? 또한, 각 채널별 전송되는 데이터가 무엇인가?	페이지 11 ~ 15
함수	DLL 이 제공하는 함수들의 기능, 인수, 반환값의 의미. 호출시 주의사항.	페이지 16 ~ 18
DLL 포함하여 작동되는 테스트용 프로젝트	사용방식의 개괄을 볼 수 있음.	VC++6.0 프로젝트 소스 : Test_LXSMWD2 개요설명 : 페이지 19 이후.

LXSMWD2 의 핵심작동방식.

응용프로그램에서는 LXSMWD2 와 연동하여 아래 그림 2 와 같은 방식으로 사용한다. 응용프로그램은 무엇보다 먼저 DLL 쪽의 장치 열기 명령을 호출하여 DLL 이 외부장치와 통신가능하게 해둔다. 그 다음 스트림시작하기 전에 관련 환경설정이 필요한 것이 있으며, 이를 실행시킨다. 그 다음 스트림 시작명령을 DLL 에 전달하면, DLL 은 장치에서 연속적으로 데이터를 받아오는 과정이 시작되며, 응용프로그램 쪽으로 알려야 할 정보를 메시지 형식으로만 전달하게된다. 이때 전달되는 메시지는 5 가지 종류가 있으며, 1 개는 측정량을 가져가라는 메시지이고, 나머지 4 개는 센서의 상태정보에 해당한다. 측정량을 가져가라는 메시지 전송시 인자로 응용프로그램이 가져가야 할 데이터가 저장되어 있는 메모리의 시작번지가 전달되며, 응용프로그램은 다음 메시지가 발생되기 전까지 해당 데이터를 전부 확보해야 한다.

스트림모드에서의 작동과정에서는 필요한 모든 데이터가 메시지라는 방식으로 전송되지만, 별도로 응용프로그램이 알기를 원하는 시점에 특정 정보를 요청할 수 있는 함수들이 있다. 이것은 주로 스트림전송중인 아닌 시점에 장치의 정보를 알고자 할 때 사용되는 함수들이며(예를들어 맥파센서가 연결되었나?), 스트림전송중이든 아니든 본 함수를 호출할 수 있다. 아래 블록도에 언급한 부분에 해당하는 함수들의 리스트를 표에 보이고 있다.

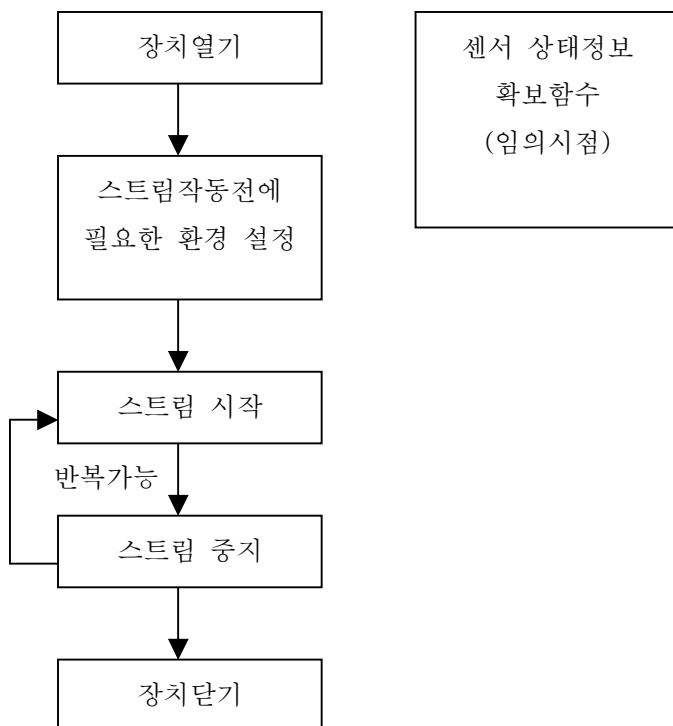


그림 2. 응용프로그램이 본 DLL 과 연동하여 사용할 때의 전체 작동방식.

기능	함수	개요
장치열기	Open_Device(int pid);	외부장치의 고유아이디
스트림작동전에 필요한 환경설정.	SetConfig_Msg(int msgtype_idx, HWND hwnd_msgtarget int msgid, unsigned char on_off)	메시지타입: 0,1,2,3,4,5 윈도우 핸들: 메시지를 받을 응용프로그램측의 윈도우 핸들. 메시지아이디: 메시지 전송시 사용할 메시지 아이디 지정. 메시지온오프: 메시지를 전송의 on off 설정.
	SetCount_ReturnData(short num_count);	단위 수량데이터의 정수배로 지정가능.
	Select_Channel(unsigned char * is_select_channel);	DLL 이 전송하는 여러 채널 중에서 임의 채널 선택.
스트림 시작	Start_Stream();	스트림데이터 전송시작.
스트림 중지	Stop_Stream();	스트림데이터 전송중지.
장치닫기	Close_Device();	장치 닫기.

기능	함수	개요
센서상태정보 확보함수.	GetInfo_PPGSensorStatus (char * connect_malfunction, char * subject);	맥파센서의 상태를 확보한다. 연결상태,, 피검체 상태.

센서의 상태정보는 스트림이 시작된 경우에는 상태가 변경된 경우에만 메시지가 발생된다. 이에 반해 상기 함수들은 임의시점에 현재 상태를 확인할 수 있는 함수다.

스트림상태에서 DLL 이 발생하는 메시지.

응용프로그램이 DLL 측에 Start_Stream()함수를 호출하여 일단 스트림을 시작시키게되면, DLL 은 메시지만으로 응용프로그램측에 정보를 제공하게 된다. 이때 총 5 종류의 메시지가 발생할 수 있으며, 각 메시지의 타입인덱스 별로 아래표에 설명 보인다.

메시지의 타입인덱스	메시지 발생사유	WParam 값 (메시지전송시 첫 번째 인자)	Lparam 값 (메시지전송시 2 번째 인자)	DLL 측의 메시지전송함수.
0	스트림데이터를 전부 확보하여 이 데이터를 응용프로그램이 가져가야 하는 시점.	없음.	스트림데이터가 저장된 float 형 배열의 시작주소.	SendMessage
1	맥파의 피크검출된 시점에 발생함.	밀리초 단위의 맥파시간격	분당맥박수	PostMessage
2	맥파센서의 연결 상태정보	없음	1, -1, -2, 0	PostMessage
3	맥파센서에 피검체 배치 정보	없음	1, -1, -2, 0	PostMessage
5	스트림 수행도중 장치 연결해제된 경우	없음.	없음.	Postmessage

타입인덱스 2,3 은 센서의 상태정보를 메시지로 전달하는 것인데, 메시지의 발생은 해당 상태가 변경된 경우에만 메시지가 전송된다. 만일 해당 상태에 대하여 메시지 전송과 무관하게 알고 싶은 경우에는 센서상태정보 확보폴링함수 GetInfo_PPGSensorStatus 를 호출하여야 해당정보를 확보한다.

응용프로그램은 상기 메시지들이 발생하도록 할 것인지 말 것인지를 제어할 수 있게 되어있다. 이는 함수 SetConfig_Msg 를 통하여 가능하다. 특정메시지 타입인덱스의 메시지 발생을 on off 시킬 수 있다.

메시지타입인덱스 2 : 맥파센서의 연결상태정보

본 메시지의 인수로 전달된 값에 따라서 아래와 같은 센서의 상태를 의미한다.

전달 값	맥파센서의 연결상태.	프로그램측의 고려사항	대응 사용자가 할일.
1	맥파센서가 연결되었다.		
0	맥파센서가 연결되지 않았다.	사용자에게 맥파센서를 연결하라는 지시를 한다. 정상검사진행불가상태. 보여야 할 문구 "맥파센서가 연결되어 있지 않습니다. 연결해주세요."	맥파센서를 연결한다.

주의: 위 기능은 ubpulse T1 에서만 지원됨.

메시지타입인덱스 3 : 맥파센서에 손가락 배치 상태정보

본 메시지의 인수로 전달된 값에 따라서 아래와 같은 의미가 있다. 아래 정보를 기반으로 프로그램의 go don't go 에 사용 하지는 말 것.

전달된 값	피검체 상태.	프로그램측의 대응 고려사항	사용자가 할일.
1	센서에 피검체가 정상 배치되었음.	센서내에 피검체가 있음을 표시	
-1	센서내에 아무것도 없다.	“센서에 아무것도 없음을 표시”을 표시한다.	손끝이 센서의 끝까지 닿도록 배치한다. 상기와 같이 하였음에도 문제가 해결되지 않는 경우(손가락이 가늘거나, 피부가 투명하여 그렇다.) 엄지손가락에 센서 배치한다..
-2	센서 내에 불투명한 것이 들어있다.	“센서에 불투명한 것이 들어와 있음”을 표시	해당물체를 제거한다.
0	메시지인덱스의 상태 - 1 이나, 0 인 경우다.	메시지인덱스 2 의 상태- 1 이나, 0 의 고려사항을 따른다.	

주의: 위 기능은 ubpulse T1 에서만 지원됨.

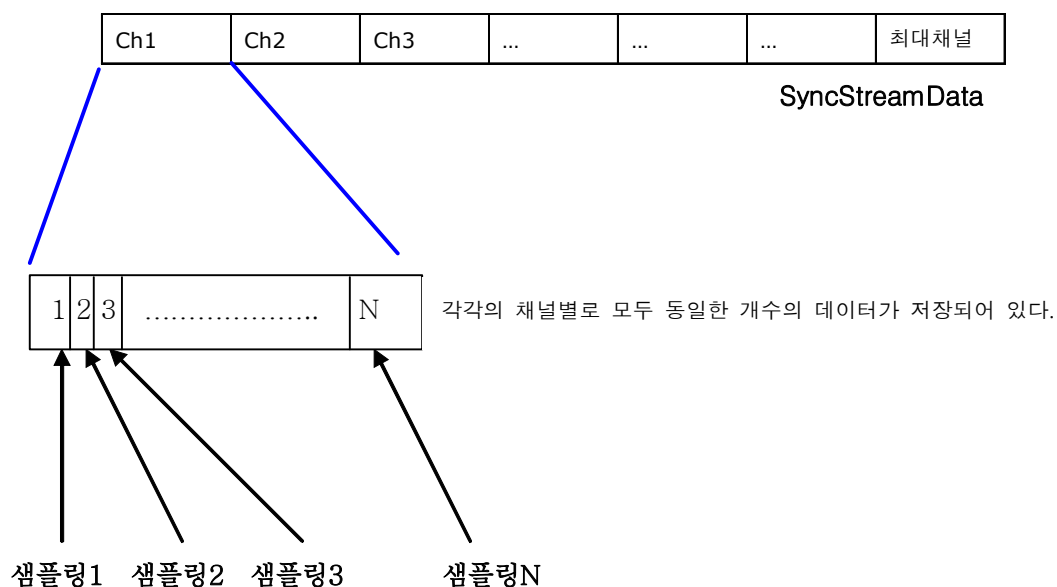
메시지타입인덱스 5 : 스트림실행 중 장치연결 해제

메시지 타입인덱스 5 가 발생된 경우, DLL 내부적으로 Stop_Stream(), Close_Device()가 내부적으로 실행되어
있는 상태이므로, 메인프로그램은 DLL 측으로 별도의 명령을 보낼 필요 없으며, 메인프로그램쪽의 오류방지
처리를 수행하면 된다.

스트림데이터의 구조

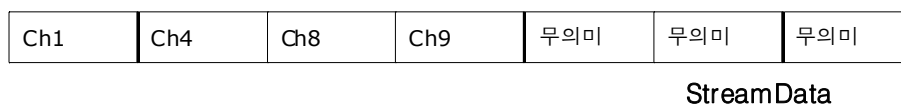
스트림 전송 상태에서 메시지타입인덱스 0은 데이터를 가져가라는 의미이며, 해당메시지의 2 번째 인수에 스트림데이터를 저장하고 있는 float 형 배열의 시작주소가 전달된다. 이때 배열에 데이터가 저장되어 있는 모습을 본 장에서 설명한다.

아래 그림에 배열에 저장되어 있는 데이터의 모습을 보이고 있다. 여기서 ch1 부터 최대채널까지 값이 기록되어있게 되며, LXSMWD2 는 최대 채널이 11 까지 있다. 각각의 채널이 어떤 의미를 갖고 있는지는 표에 보이고 있다.



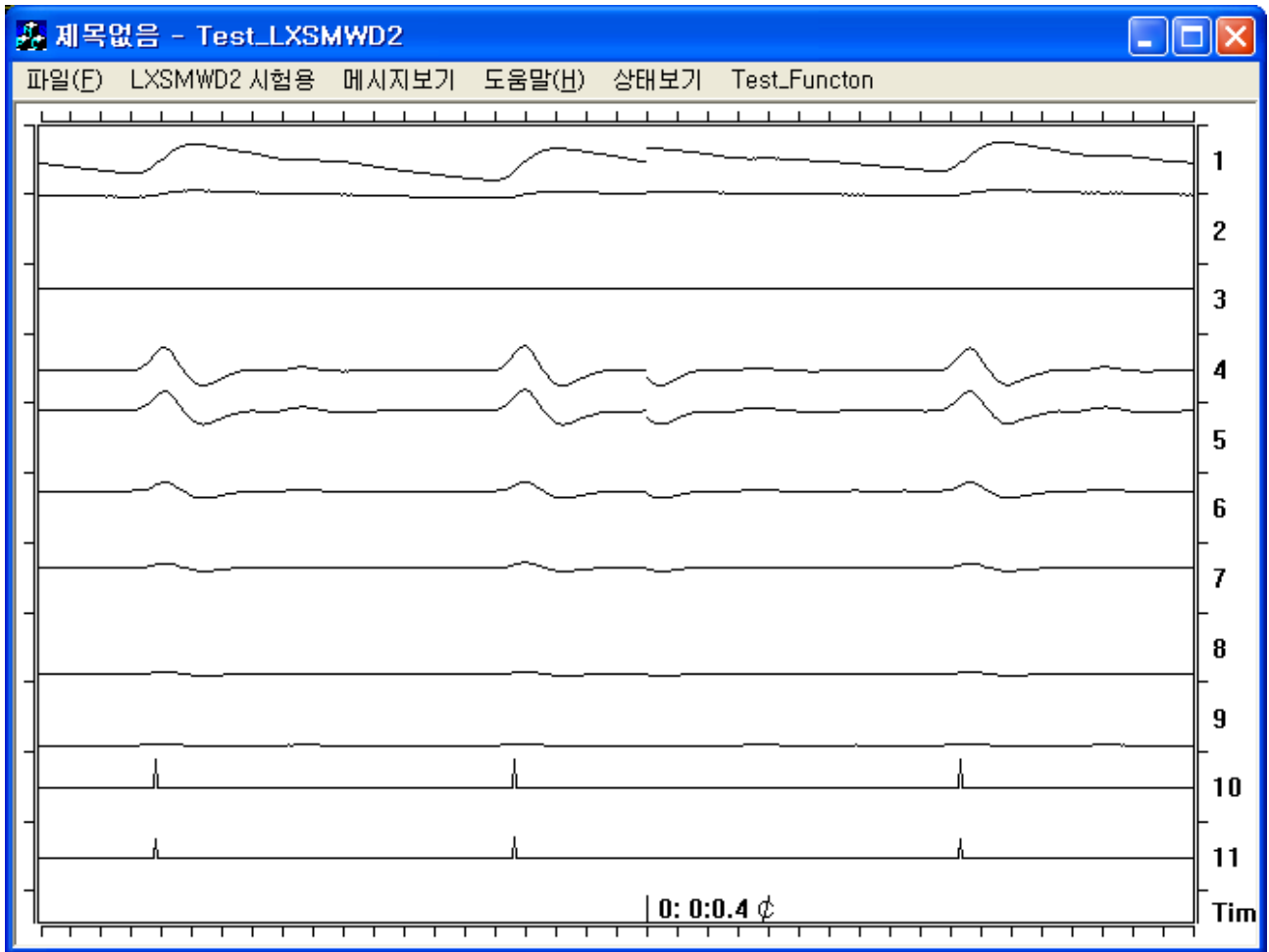
채널을 부분적으로 선택한 경우의 배열의 데이터구조.

응용프로그램에서는 스트림데이터가 보유한 총 채널 11 개를 전부 받을 수도 있고, SetConfig_Channel 이라는 함수를 이용하여 받기를 원하는 채널만 선택할 수 있다. 예를들어 이함수를 이용하여 ch1, 4, 8, 9 를 선택하였다면, 스트림데이터 배열에 값이 기록되는 방식은 아래 그림과 같이, 배열의 순서가 채널 1, 4, 8, 9 순으로 배치된다.



채널 인덱스	채널번호	측정량	설명
0	Ch1	맥파	맥파파형.
1	Ch2	-	-
2	Ch3	-	-
3	Ch4	가속도맥파	채널인덱스 0 맥파 AC 성분을 하드웨어적으로 2 번 미분한 파형.
4	Ch5	-	-
5	Ch6	-	-
6	Ch7	-	-
7	Ch8	-	-
8	Ch9	-	-
9	Ch10	맥파피크 발생시점	<p>맥파피크가 발생한 시점에 1 포인트만 1.0 이 기록되고, 나머지 구간은 전부 0.0 이 기록된다.</p> <p>(하드웨어적으로 검출된 피크 정보를 DLL 에서 피크의 시간격이 0.2 초와 3 초 이내인 경우에만 맥파피크로 인정하면서 값을 해당채널에 기록한다.)</p> <p>주의사항!!! : 스트림 전송시 맥파 피크 발생 메시지를 받기 위해서는 본 채널을 선택해야만 한다. 또한, 본 채널이 선택되면 아래 11 은 항상같이 선택된다.</p>
10	Ch11	맥파피크 시간격	<p>상기 Ch10 에서 발생한 피크와 동일한 지점에 직전피크와의 시간격이 초단위로 float 형으로 기록되어 있다. 피크 발생지점이 아닌 곳은 전부 0 이 기록되어 있다.</p> <p>(시간격이 0.2 초와 3 초이내만 기록된다.)</p> <p>주의사항!!! : ch10 이 선택되면 본채널은 항상 선택된다. Ch10 을 선택하지 않으면, 본 채널도 선택되지 않은 것으로 된다..</p>

상기 표의 파형을 아래 그림에 예로 보이고 있다.



스트림데이터 관련 코딩 상세설명.

스트림데이터의 구조를 보면, 각각의 채널별로 1 개의 샘플링 데이터만 있는것이 아니라, "최소수량의 정수배"의 샘플링 데이터가 있고, 최소수량은 4 이며 배수값은 SetCount_ReturnData()라는 함수로 지정가능하다.

만일 배수값을 1로 하면, 스트림데이터의 각 채널에는 모두 4 개의 샘플링 데이터가 있게된다.

즉, 채널 1 구간에 채널 1 의 4 개 샘플링 데이터가 배치되어있고, 그다음 채널 2 의 4 개 샘플링데이터가 배치된다. 따라서, 각각의 채널별로 4 개씩 데이터가 있음을 고려하여 데이터를 확보해야한다.

메시지로 전달된 인자로부터 float 형 배열로 데이터를 받는 방법.

LL 내에 float 형 배열이 있고, DLL에서는 실시간으로 수집되는 데이터를 이열에 저장 한다. 실시간 데이터 수집과정에서 DLL은 정해진 수량만큼 데이터가 확보된 시점에 메시지를 발생하며, 이를 수신한 프로그램측 메시지핸들러 함수 내에서는 데이터를 이용하는 과정이 이뤄져야 한다. 메시지로 전달되면서 데이터에 접근하기 위한 정보는 메시지 인자 중 IParam에 그 정보가 있다. IParam으로 DLL의 배열의 포인터가 전달된다. 예를들어 array[200]이라는 배열이 있다면, array에 해당하는 것을 전송한다. 따라서, 우리는 IParam을 이용하여 활용하기 위해서는 IParam을 float 형 포인터로 형변환을 하고 (float *)IParam 이것이 마치 array인것 처럼 활용하면된다. 구체적인 코드예를 이용하여 설명해보자.

우리의 응용프로그램에서 데이터를 받아둘 배열을 1개 만들자.

```
float save_data[16];
```

배열의 크기결정을 위해서는 아래식을 이용해야 한다.

배열의 최소크기 = 총채널수 × 각채널당 샘플링수량 (총채널수 = 4, 샘플링수량 = 4 이므로 배열의 크기는 16 이 나온다)

데이터 확보하는 전용함수를 선언하고, 함수인자로 float 형 포인터변수를 전달하자.

```
void Get_Data(float * my_data)
{
    for(int i = 0 ; i< 16 ;i++)
    {
        save_data[i] = *(my_data+i); // my_data 의 전체 데이터를 save_data 로 이전했다.
    }
}
```

위의 함수가 호출될 장소는 우리의 메시지핸들러인 OnStreamData 내에 두면된다.

즉,

```
void CChildView::OnStreamData(WPARAM wParam, LPARAM lParam )
```

```
{
```

```
Get_Data((float *) (lParam)); // DLL 속의 데이터를 우리가 사용가능한 배열로 이전하는 함수
```

```
}
```

save_data[i] 의 데이터 배치상황.

채널 1의 샘플링데이터 1, 채널 1의 샘플링 데이터 2, 채널 1의 샘플링 데이터 3, 채널 1의 샘플링 데이터 4,

채널 2의 샘플링데이터 1, 채널 2의 샘플링 데이터 2, 채널 2의 샘플링 데이터 3, 채널 2의 샘플링 데이터 4,

채널 3의 샘플링데이터 1, 채널 3의 샘플링 데이터 2, 채널 3의 샘플링 데이터 3, 채널 3의 샘플링 데이터 4,

채널 4의 샘플링데이터 1, 채널 4의 샘플링 데이터 2, 채널 4의 샘플링 데이터 3, 채널 4의 샘플링 데이터 4,

함수설명서

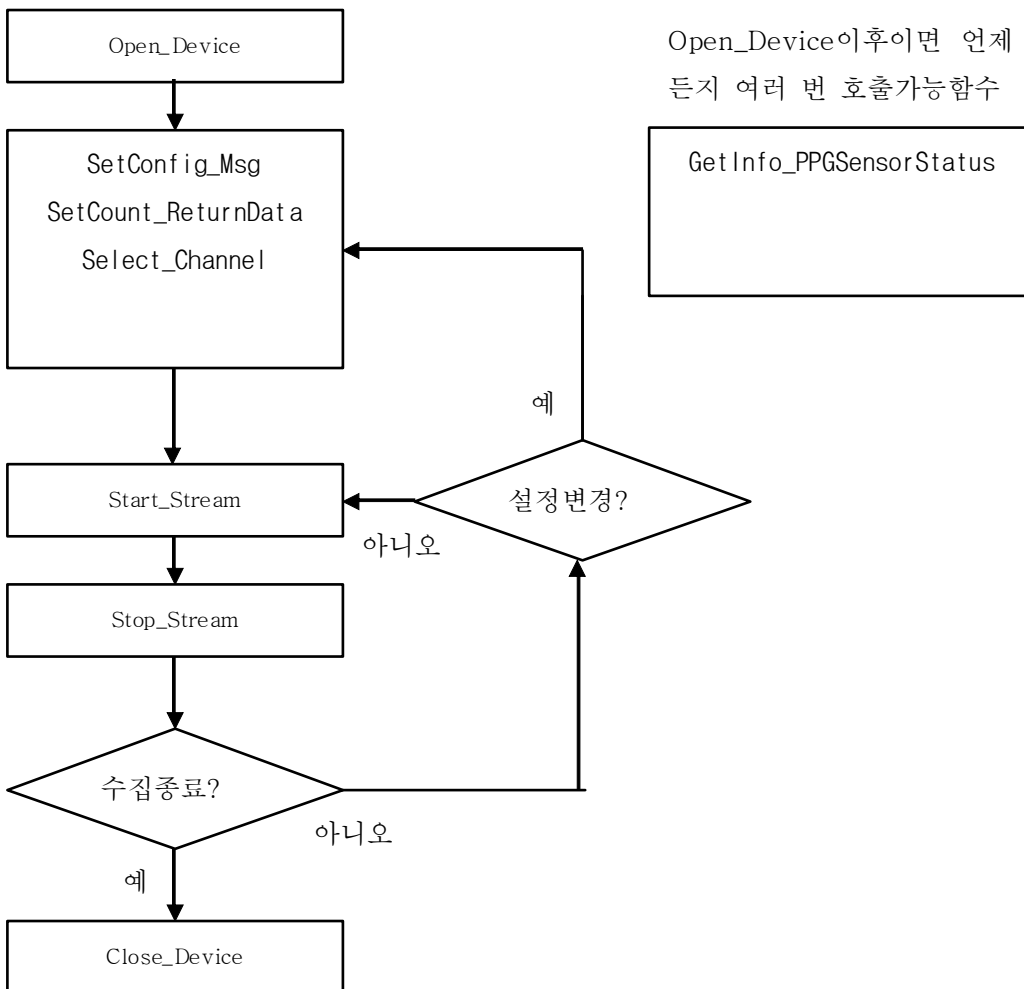
Dll 이 제공하는 전체함수리스트.

함수	설명	오류반환값
Open_Device(int pid);	원하는 장치를 opne 한다. pid 에는 장치의 고유아이디를 전달한다. ubpulse T1 pid = 27(십진수), RP920 pid = 41(십진수)	-2: 중복호출한 경우. -1: 장치해들 못 찾은 경우.
Close_Device();	열었던 장치를 닫는다.	-1: 장치열지도 않고 호출된 경우
Start_Stream();	스트림 전송을 시작시킨다. 스트림 전송이 시작되면 dll 은 메시지로 정보를 제공함.	-1: Open_device 하지 않고 호출한경우. -3: 장치명령전달실패
Stop_Stream();	스트림 전송을 중지한다. 메시지지 전송도 중단된다.	-1: Open_device 하지 않고 호출한 경우. -2: 스트림중이 아닌네, 호출한 경우.
SetConfig_Msg(int msgtype_idx, HWND hwnd,msgtarget, int msgid, unsigned char on_off);	스트림전송시 총 6 타입의 메시지를 발생하게 되는데, 각타입별 메시지받을 윈도우 핸들과 메시지 아이디를 지정해 주는 것이다.	-1: 인덱스범위가틀렸다.(0~5) -2: 메시지아이디범위틀렸다. (WM_USER+1 ~ WM_USER+32767)
SetCount_ReturnData(short num_count);	스트림 전송시 dll 은 기본적으로 각채널별로 4 샘플의 데이터를 전송하게 된다. 메인프로그램에서 4 샘플의 몇 배수로 받을 것인지를 지정한다. 이때	-1: num_count 범위가 틀렸다. (64 이하만 허용) -2: 스레드수행중에 호출한

	배수는 1,2,4,8,16,32,... 로 설정해야 한다.	경우.
Select_Channel(unsigned char * is_select_channel);	Dll 은 총 11 채널의 정보를 갖고 있다. 이 중에서 받고 싶은 채널만 선별하는 함수다.	-2: 스레드수행중에 호출한 경우.
Short GetInfo_PPGSensorStatus(char * connect_malfunction, char * subject);	맥파센서의 상태정보를 받을 수 있는 함수다. 인자로 값을 반환해준다. Connect_malfunction 의 값은 페이지 메시지타입인덱스 2 의 전달된 값과 동일. Subject 의 값은 메시지타입인덱스 3 의 전달된 값과 동일.	-1 : Open_device 하지 않고 호출한 경우.

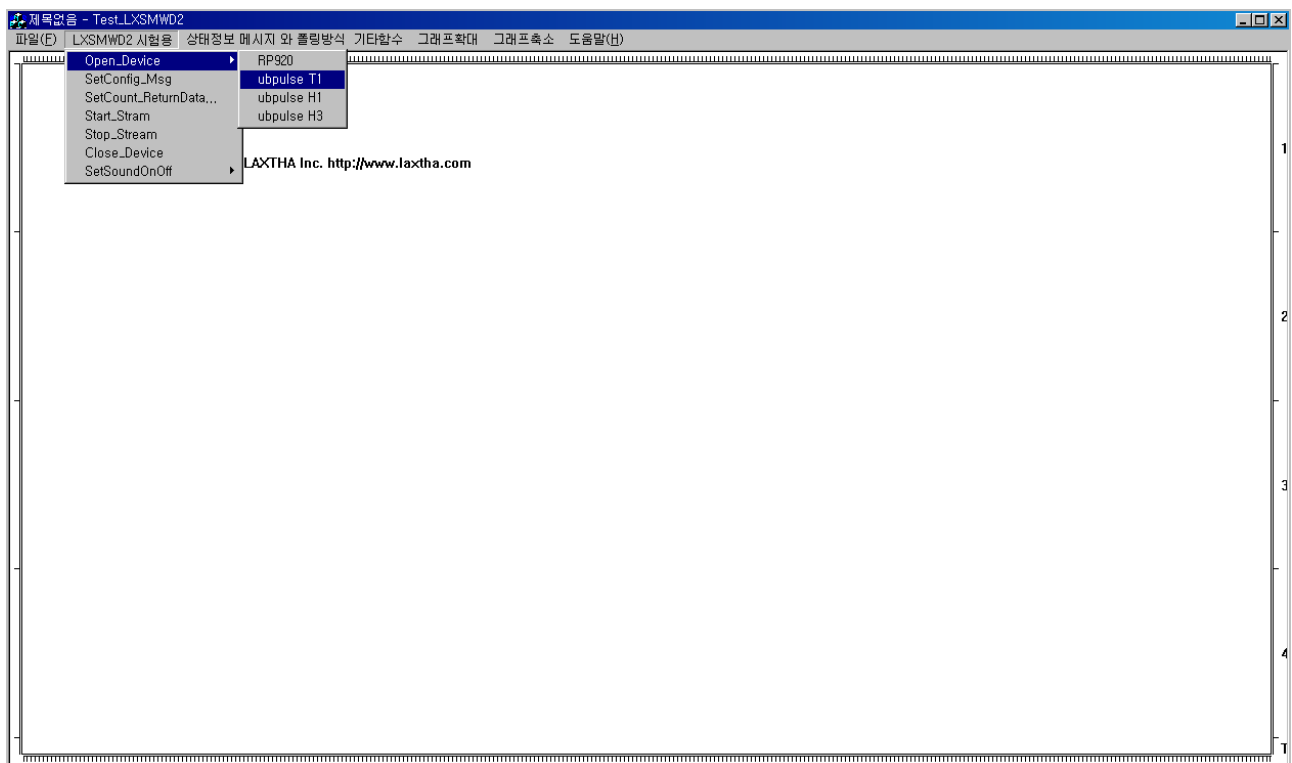
함수호출순서

함수들간의 호출순서는 아래 순서로만 사용한다. 이 중에서 센서상태정보를 확보하는 함수는 Open_Device 호출된 이후이면 언제든지 호출 가능하다.



LXSMWD2 활용 예제 VC++ 6.0 프로젝트

DLL LXSMWD2 를 활용하는 예제 프로젝트 소스로서 Visual C++ 6.0 으로 제작된 Test_LXSMWD2 가 제공되고 있으며, 이 프로젝트에서는 DLL 에서 제공하는 함수와 메시지를 사용하는 예를 보이고 있다. 실행화면은 아래 그림과 같다.



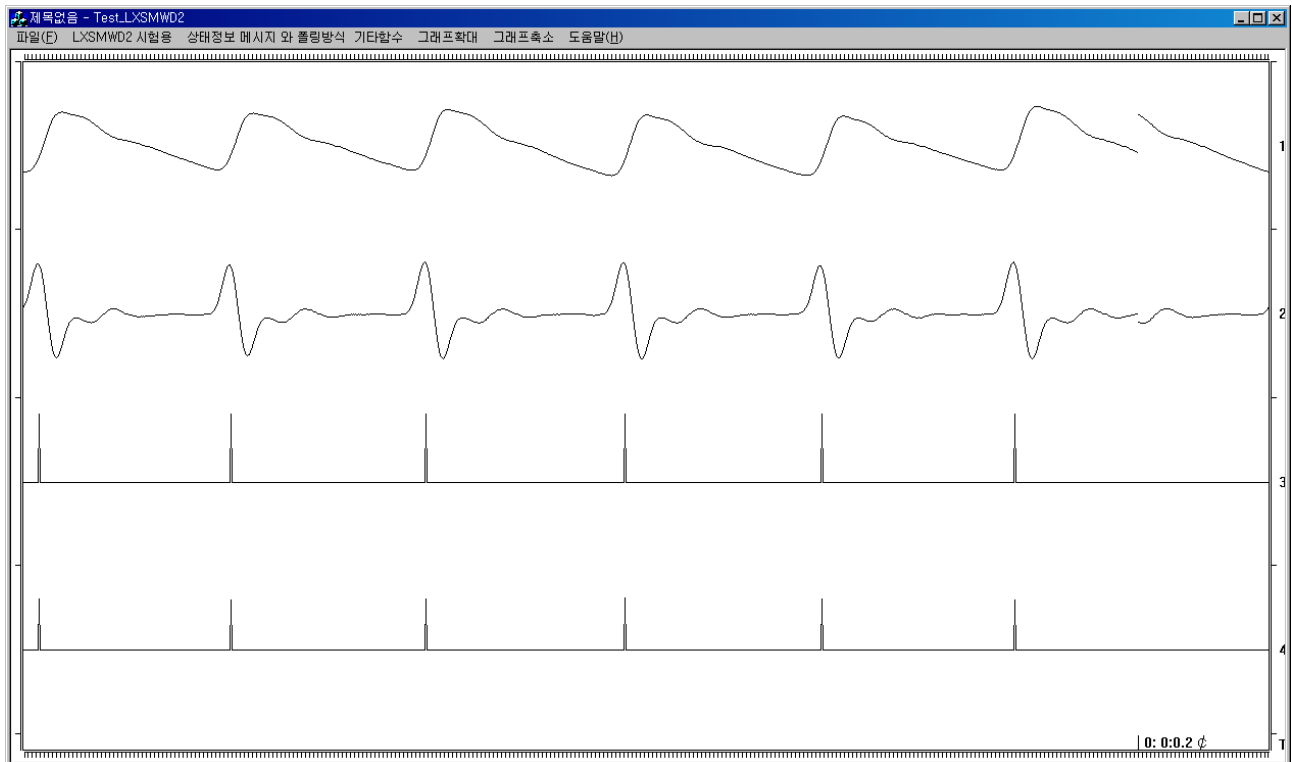
화면 1 – Test_LXSMWD2 실행화면.

Test_LXSMWD2 사용법.

1. 장치를 컴퓨터에 연결하고, Test_LXSMWD2.exe 를 실행한다.
2. 상단메뉴에서 LXSMWD2 시험용 클릭 -> Open_Device -> 컴퓨터에 연결된 장치를 선택 클릭한다.
Open_Device() 함수가 실행된다.
3. 상단메뉴에서 LXSMWD2 시험용 클릭 -> SetConfig_Msg 클릭.
4. 상단메뉴에서 LXSMWD2 시험용 클릭 -> SetCount_ReturnData... 클릭 -> 설정 대화상자에서 배수 1 을 선택하고 확인.

5. 상단메뉴에서 LXSMWD2 시험용 클릭 -> StartStream 클릭.

StartStream 을 실행하면 장치로부터 실시간으로 데이터 전송이 시작되며, 데이터가 화면에 디스플레이 된다. 아래 디스플레이되는 화면을 보이고 있다. 위에서부터 1 번 파형은 맥파 파형에 해당하고 2 번 파형은 맥파파형을 2 번 미분한 파형을 보이고 있고, 3 번째 데이터는 심박 발생시점에 1.0 의 값이 기록되어 전송되며, 4 번째 데이터는 심박시간격 을 초단위로 전송중인 모습이다.



6. 상단메뉴에서 LXSMWD2 시험용 클릭 -> StopStream 클릭.

스트림 전송이 중지된다. 이 단계에서 스트림 전송을 재개하려면 StartStream 클릭한다.

7. 상단메뉴에서 LXSMWD2 시험용 클릭 -> CloseDevice 클릭.

장치와의 연결을 끊는다. 이 단계에서 다시 장치와 연결하여 통신하기 위해서는 2 번부터 다시 시작한다.

심박수와 심박 시간격을 메시지로 수신하는 기능시험.

스트림 데이터로 전송되는 파형 및 심박시점 정보와 심박 시간격 정보보다는 심박 시간격과 분당맥박수를 메시지 형태로 전송해주는 기능을 이용하는 것이 더 유용하다. Test_LXSMWD2 에서 이를 시험하기 위해서는

스트림전송이 실행중인 상태에서 (앞의 설명 5 번 단계)에서 상단메뉴 -> 상태정보메시지와 폴링방식 기타함수 클릭 -> PPG Pulse, Status.. 클릭한다. 그러면 아래 대화 상자가 활성화 된다.

대화상자에서 메시시 타겟을 여기로 전송명령 버튼을 클릭한다. 그럼 펄스 시간격과 분당맥박수에 심박이 발생할때마다 신규 데이터로 갱신된다. 센서연결/장착/고장상태 역시 장치의 상태를 반영하여 보여주게 된다.

주의사항 – 센서연결/장착/고장상태 는 ubpulse 시리즈 에서만 지원되는 기능임.

The image shows a Windows-style dialog box titled "PPG Pulse & Status View". It has a close button (X) in the top right corner. Below the title bar, there are three buttons: "메시지타겟을여기로 전송명령" (Send message target to here), "OK", and "Cancel".

Under the "Static" section, there are two input fields: "펄스시간격[초]" (Pulse interval [sec]) with the value "0.73", and "분당맥박수" (Pulse rate) with the value "82".

Below this, there is a section titled "센서연결/장착/고장상태" (Sensor connection/installation/fault status). Inside this section, there is a button labeled "GetInfo_PPGSensorStatus".

Below the button, there are three status indicators, each with a label and a text box:

- "센서연결상태" (Sensor connection status) with the value "연결됨" (Connected).
- "센서장착상태" (Sensor installation status) with the value "Normal Subject - 손가락 들어와 있음." (Normal Subject - Finger is inserted).
- "센서고장유무" (Sensor fault status) with the value "정상작동" (Normal operation).