

Keyword : PSL-DAQ Library dll, PSL-DAQ Visual C++ Library, PSL-iECG2, PSL-iEMG2, PSL-iEOG2

PSL-DAQ Visual C++ Library Manual

Manual

Contents

- 1 Definition of Terms**
- 2 1. PSL-DAQ Visual C++ Library Summary**
- 3 2. PSL-DAQ Visual C++ Library Description**
 - 3 2.1 Understanding of the functionality of PSL-DAQ Visual C++ Library
 - 5 2.2 Description of PSLDAQ_dll Library Function
 - 8 2.3 Description of PSLDAQ_dll user-defined message
- 9 3. Examples of PSL-DAQ Visual C++ Library**

Definition of Terms

PSL-DAQ	<p>PSL-DAQ is a small 2-channel analog signal data acquisition module released with PSL-iModule V2 series(PSL-iECG2, PSL-iEMG2, PSL-iEOG2).</p> <p>PSL-DAQ is designed to receive analog 2-channel signals and supply DC 5V power source to the sensor module through a single input port. Therefore, it can be simply connected to the sensor module without any other power supply.</p> <p>PSL-DAQ has 2-channel 16bit ADC and 32bit micom, and re-transmission protocol(PysioLab's USB communication protocol) ensures reliable transmission of the obtained data to the PC.</p> <p>PSL-DAQ provides a powerful monitoring software, and also provides the LabVIEW and Visual C++ libraries together.</p>
PSL-DAQ RMSW	PSL-DAQ RMSW is software of PSL-DAQ on PC. It is available for real-time monitoring/ storing/ reviewing data obtained from the PSL-DAQ.
PSL-iModule V2	<p>PSL-iModule V2 is the brand name of small bio-signal sensor module and is composed of ECG, EMG, and EOG module. PSL-iECG2 guarantees electrical safety through isolation power and signal during measuring. For the high quality of signal input the premium lead cable with shield is provided, and the cable is applied a stereo jack to connect simply.</p> <ul style="list-style-type: none"> - PSL-iECG2 is a small ECG module that outputs ECG waveforms and heart beats. - PSL-iEMG2 is a small EMG module that outputs EMG waveforms and EMG Evelope signal. - PSL-iEOG2 is a small EOG module that outputs EOG and EOG Direction Event signal.
PSLDAQ_DLL.dll	PSLDAQ_DLL.dll includes a function to control PSL-DAQ. Because It consists of the type of Regular DLL, it can be used in Visual C++ and LabVIEW.
PC DLL Application Program	It means that the user develop application programs by using DLL of PSL-DAQ Library.

Precaution

- PSL-DAQ Visual C++ Library is the dedicated library for the PSL-DAQ. To use the library, please connect a PSL-DAQ to your PC.
- The PSL-DAQ Visual C++ Library function can be used only when the USB driver of PSL-DAQ is installed.
- Before using PSL-DAQ Visual C++ Library, please turn on the power of the PSL-DAQ, and check if analog data can be acquired normally through the PSL-DAQ RMSW software.
- For more details, please contact PhysioLab Co., Ltd.

1. PSL-DAQ Visual C++ Library Summary

PSL-DAQ Visual C++ Library allows the use of the PSL-DAQ functions in Visual C++.

PSL-DAQ converts analogue signals input by 2 channels into digital signals (display in data henceforth) at 1,000 samples per second and transmits to computer using the USB communication method. The data of 2 channels transmitted to computer is acquired data in real time using PSL-DAQ Visual C++ Library and is displayed the data in a graph. The library is provided along with a files of examples for explanation and learning of Visual C++ function.

PSL-DAQ Visual C++ Library consists of functions to be used by users and PSL-DAQ driver. Functions that users can call are provided in PSLDAQ_DLL.dll.

PSL-DAQ Visual C++ Library is developed by using Visual Studio 2008 Professional Edition.

Main Functions of PSL-DAQ Visual C++ Library

- Acquires 2 channel data obtained from PSL-DAQ from PC DLL add-on programs.
- Checks the device ID of connected PSL-DAQ.
- Transmits messages to check information related to connection status and signal acquisition of PSL-DAQ in PC DLL add-on programs.

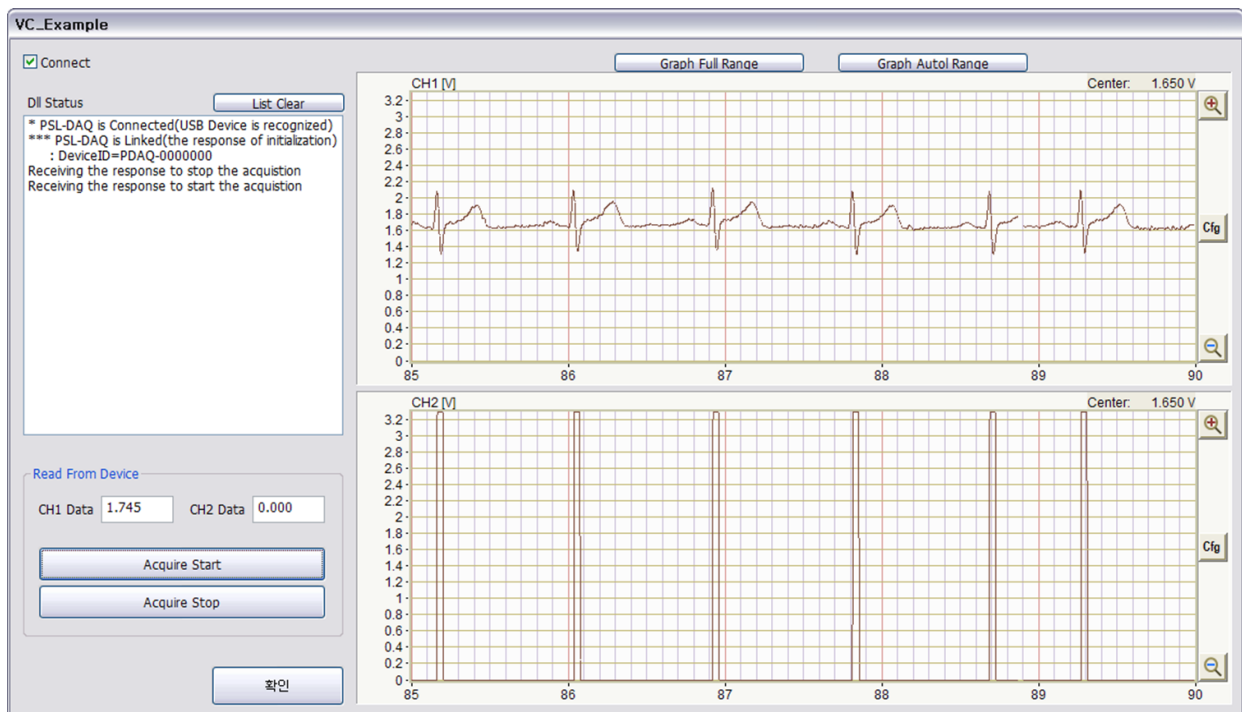


Fig. 1. ECG and Beat data acquisition example using with PSL-DAQ Visual C++ Library.

2. PSL-DAQ Visual C++ Library Description

2.1 Understanding of the Functionality of PSL-DAQ Visual C++ Library

There are basic functions that you can utilize by using PSL-DAQ Visual C++ Library - obtaining data, and checking the status of PSL-DAQ. Functionality and the relevant function will be provided in the later session.

● Data Acquisition

The input signal of PSL-DAQ is 2 channel analogue signal in the range of 0~3.3V, and signals input are converted to digital signals of 1,000 SPS (Samples per Second) through 16 bit ADC of the internal device. Digital signals are transmitted to computer by USB communication, and data can be acquired from user programs by using PSL-DAQ Visual C++ Library.

To use PSL-DAQ Visual C++ Library, you must start with the PSLDAQ_OPEN() function, complete by using the PSLDAQ_Close() function, and assign functions of your choice in between. The functions relevant to data acquisition in library are PSLDAQ_AcquireStart(), PSLDAQ_AcquireStop(), and PSLDAQ_Read().

Table 2. The library functions used for data acquisition

Functions	Description
PSLDAQ_Open()	Activates the communication connection function to PSL-DAQ, and provides overall initialization, memory allocation, and device handling to operate PSLDAQ_DLL.dll. If a multiple numbers of PSL-DAQ are connected to your computer, the PSLDAQ_Open function must be used according to the number of devices being connected and obtain device handle for each device.
PSLDAQ_Close()	Performs finishing process, such as disable communication connection of devices and removal of allocated memories
PSLDAQ_AcquireStart()	Starts data acquisition
PSLDAQ_AcquireStop()	Stops data acquisition
PSLDAQ_Read()	Reads acquired data

Note) The process description of data delivery and reading is provided in PC DLL add-on programs

- The data of Queue (Out Queue and internal device transmission Queue) for the internal communication of PSL-DAQ is stored in Receive Queue of DLL through USB communication. (Since communication error may occur when the data of internal device Send Queue is transmitted to the receive Queue, the communication function to remove errors is performed in DLL and the FW of PSL-DAQ.)
- Read the Receive Queue data of PSLDAQ_DLL.dll using the PSLDAQ_Read() functions that process WM_RECEIVE_DATA messages, and the value of WPARM is 0.
- Information, such as problems caused by Queue and communication during data transmission are also transmitted using WM_RECEIVE_DATA messages, and the value of WPARM has a non-zero value.

● Checking of the status of PSL-DAQ

The status check of PSL-DAQ is performed in order of initialization through the PSLDAQ_Open() function, receive device status information, and ending.

The information of PSL-DAQ status is transmitted by a user-defined message of WM_RECEIVE_DATA.

A device name(ID) can be obtained through the function of PSLDAQ_GetDeviceID(), and it can be called at any time after calling the function of PSLDAQ_Open().

Transmission message information about the status of PSL-DAQ

◦ The message information of WM_RECEIVE_DATA

- Information about communication link of PSL-DAQ

(Due to the characteristics of USB devices, connection can be disconnected at a certain point after the PSLDAQ_Open() function is called; thus, information is transmitted every time connection status is varied.)

- Responses to data acquisition start/stop request
- Status information of Queues relevant to data acquisition
- Error information of communication about the received packets

Note) Communication Connection

PSL-DAQ is a device that receives power and implements communication via USB and it can be connected or removed from computers at anytime. When power is supplied to PSL-DAQ (when the power is turned on and connected to the USB port, your computer will be able to control PSL-DAQ through communication. This state is described as active communication, while the opposite state is inactive communication.

Of the PSLDAQ_Open() functions, there is a communication connection function activation. At the time of function calling, devices cannot be controlled (communicated with PC) if power is not supplied to PSL-DAQ. Since communication can be connected when power is supplied to PSL-DAQ after the function is called, such a state is described as function activation, meaning that connection is feasible anytime. Also, the expression "function activation" is used because a sustainable function is required to detect changes in communication since the connection may vary even after connection. For more information about changes in the status of communication connection, see user-defined message (WM_RECEIVE_DATA).

2.2 Description of PSLDAQ_DLL.dll Library function

● PSLDAQ_Open()

Role		<ul style="list-style-type: none"> • Initializing memory allocation/thread creation related to DLL - A function that can be connected to PSL-DAQ will be activated (Communication connection activation). - A essential function to use PSL-DAQ. - Create device handle to be used in functions provided by DLL .
Function Description	Parameter	<ul style="list-style-type: none"> • HWND MsgWnd <ul style="list-style-type: none"> - A window handle that processes WM_RECEIVE_DATA message transmitted from DLL - MsgWnd must be recorded. • char *DeviceName <ul style="list-style-type: none"> - It is the unique device name(ID) of the PSL-DAQ and is called by PSLDAQ_GetDeviceName(). If several devices are connected to the pc at the same time, also it can be identified in Device Selection Dialog. The format of ID is PDAQ-xxxxxxx, and x is a number. - If several devices are connected and an empty string is entered in DeviceID, the Device Selection Dialog will appear. If device ID is recorded in DeviceID, connection will be initiated with the PSL-DAQ of the ID entered. • unsigned int *DeviceHandle <ul style="list-style-type: none"> - Device handle, used in all functions of DLL except for the PSLDAQ_Open() function. In order to handle several devices, the number of PSLDAQ_Open() functions according to the devices must be called. each DeviceHandle has unique value.
	Return	<ul style="list-style-type: none"> • 0 : The PSLDAQ_Open() function works normally. • 1 : The PSLDAQ_Open() function works abnormally. (Returns 1 when this function is called without installing USB driver for PSL-DAQ).
Note		<ul style="list-style-type: none"> • The PSLDAQ_Open() and the PSLDAQ_Close() function should be used together. • Devices will not be opened if the parameter of DeviceName does not match with the name(ID) of PSL-DAQ. • To use functions, check that devices work normally by using PSL-DAQ RMSW.

● PSLDAQ_Close

Role		<ul style="list-style-type: none"> Removing the memories and threads allocated by PSLDAQ_Open() - Finishing DLL function(disconnecting communications and so on)
Function Description	Parameter	• unsigned int DeviceHandle : Device Handle
	Return	-
Note		<ul style="list-style-type: none"> If the PSLDAQ_Open() function is used, the PSLDAQ_Close() function must be called. Calls PSLDAQ_Close after all functions are used. After using the PSLDAQ_Close() function, the PSLDAQ_Open() function can be called again.

● PSLDAQ_AcquireStart

Role		• Starting data acquisition from PSL-DAQ
Function Description	Parameter	• unsigned int DeviceHandle : Device Handle
	Return	-
Note		• It operates normally when PSL-DAQ is linked.

● PSLDAQ_AcquireStop

Role		• Stopping data acquisition from PSL-DAQ
Function Description	Parameter	• unsigned int DeviceHandle : Device Handle
	Return	-
Note		• It operates normally when PSL-DAQ is linked.

● PSLDAQ_Read

Role		<ul style="list-style-type: none"> Reading acquired data from PSL-DAQ.
Function Description	Parameter	<ul style="list-style-type: none"> unsigned int DeviceHandle : Device Handle double *pdData <ul style="list-style-type: none"> The array pointer is 1 sample of the floating point data obtained from PSL-DAQ, the size of array is double [2]. INT16 *pIDData <ul style="list-style-type: none"> The array pointer is 1 sample of the floating point data obtained from PSL-DAQ, the size of array is INT16 [2]. It is not necessary to be used.
	Return	<ul style="list-style-type: none"> 0 : No data is read. The value of pdData can't be used. 1 : Data is read. The value of pdData can be used.
Note		<ul style="list-style-type: none"> It operates normally when PSL-DAQ is linked. PSLDAQ_AcquireStart function is called first.

● PSLDAQ_GetDeviceID

Role		<ul style="list-style-type: none"> Obtaining a unique name(ID) from the linked PSL-DAQ devices.
Function Description	Parameter	<ul style="list-style-type: none"> unsigned int DeviceHandle : Device Handle char *DeviceID <ul style="list-style-type: none"> DeviceID points char [256] array, it includes a unique device name(ID) of the opened PSL-DAQ. The format of Name(ID) is PDAQ-xxxxxxx, and x is a number.
	Return	-
Note		<ul style="list-style-type: none"> It operates normally when PSL-DAQ is linked.

2.3 Description of user defined message in PSLDAQ_DLL.dll

User-defined message are posted in order to transmit a variety of information via PC DLL add-on programs in PSLDAQ_DLL.dll. There is used one of user defined message(WM_USER+1, WM_RECEIVE_DATA) in the PSLDAQ_DLL.dll, and it is posted to the PC DLL add-on program.

The type of message information is divided by WPARM of value. If WPARM is not zero of value, the informations are relevant to the event of communication and acquisition. If WPARM is zero of value, it means that the received data is existed.

● WM_RECEIVE_DATA

Role	<ul style="list-style-type: none"> The message is posted when it becomes necessary to provide information related to data acquisition or communication link of PC DLL add-on programs with PSL-DAQ.
Description	<ul style="list-style-type: none"> WPARM of value is not 0 : Event information of communication and acquisition <ul style="list-style-type: none"> Informations related to the acquisition of data <ul style="list-style-type: none"> LP_D2P_DATA_START : Receiving a response about the request for starting acquisition of data. LP_D2P_DATA_STOP : Receiving a response about the request for stopping acquisition of data. LP_DEV_BUF_OVERFLOW : Receiving a Send Queue overflow of PSL-DAQ. LP_PC_QUEUE_OVERFLOW : Receiving a overflow of internal receive Queue in the PSLDAQ_DLL.dll. LP_RCV_PACKET_LENGTH_ERROR: Receiving a packet length error of the received data from PSLDAQ_DLL.dll. Informations related to the status of communication with PSL-DAQ. <ul style="list-style-type: none"> LP_DEV_OPEN : The USB device is recognized. No processing is required by the user. LP_COMM_INIT : After opening the USB device, the communication is established properly. LP_DEV_CLOSE : The USB device is disconnected. The connection of communication is terminated. WPARM of value is 0 : Notification related to the receiving data. <ul style="list-style-type: none"> LP_D2P_DATA_IN : It notifies that the acquired data has been received. It means the PSLDAQ_Read() function is available to read the data.

3. Examples of PSL-DAQ Visual C++ Library

Before developing PC DLL add-on programs and running PSL-DAQ Visual C++ Library examples, it is recommended that you check if your computer and PSL-DAQ KIT are working normally through PSL-DAQ RMSW software. Detailed notes are provided in the example source code, and additional explanations are omitted as interpretation of examples are determined possible by referring to the provided manual.

Implements for developing PC DLL add-on programs

- With the use of library functions, use the `PSLDAQ_Open()` function at the time you wish to link with PSL-DAQ.
- Use the `PSLDAQ_Close()` function at the point when the use of library functions are no longer necessary.
- Implement the function (it is the `ReceiveData()` function in the example) to process the `WM_RECEIVE_DATA` message delivered in dll.
- Afterwards, use the remaining functions provided in the library and the information of the message sent from DLL to develop programs.

● Hardware Preparation

Before you use the library to work programming, check the prepared hardware is working properly with the PSL-DAQ RMSW software on your PC, as shown in Figure 2.

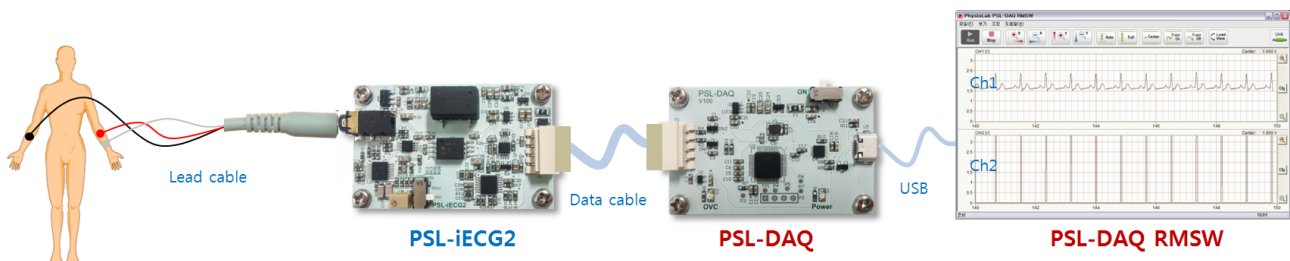


Figure 2. Preparation of hardware for the example of PSL-DAQ Visual C++ Library

● Example execution screen

Figure 3 is a screen that execute provided source code. If PSL-DAQ is linked, it can be executed to acquire data and the Dll Status shows the status of PSL-DAQ.

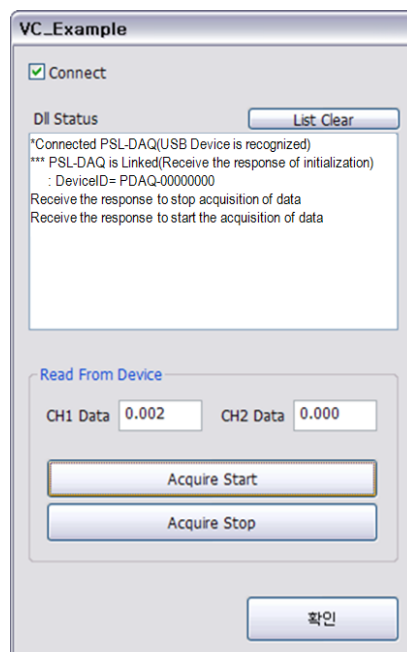


Figure 3. The screen that executed the example source code.

Fig. 7 is an additional screen that displays acquired data in a graph in order to help visual understanding. The source code is excluded with the graph display, which require users to develop according to application programs. For user convenience, however, an execution file ("[execution file compilation]_Example with Graph.exe") is provided.

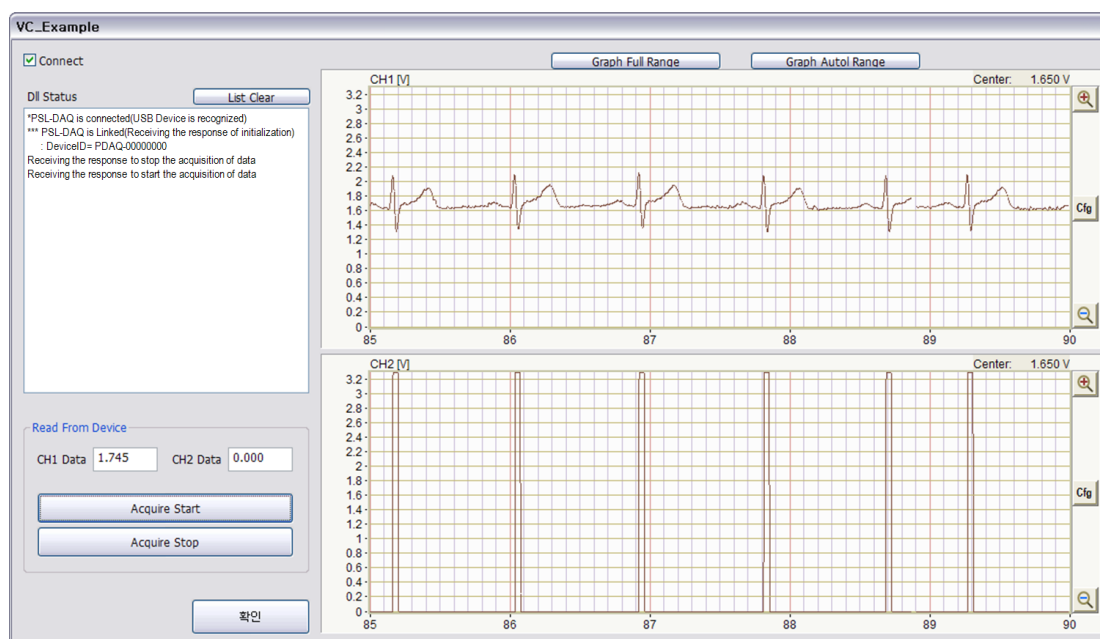


Figure 4. A screen that added the graphic display function.

Homepage : <http://www.physiolab.co.kr> E-mail : physiolab@physiolab.co.kr
Offline Shop : PhysioLab Co., Ltd.
Online Shop : PhysioLab <http://storefarm.naver.com/physiolab>
Device Mart <http://www.devicemart.co.kr/goods/brand.php?seq=1308>
Tel. +82-51-325-2868, Fax. +82-51-325-2869

Copyright © 2015, PhysioLab Co., Ltd.