Neurotechnology Research Systems

PlexStim™ Electrical Stimulator DLL Guide

www.plexon.com

PlexStim™ Electrical Stimulator DLL Guide

Contents

Documentation History

Date	Version	Notes	Author
May 2015	STMTN0002a	- Initial Creation	Tanya Mukhina Stacie Hyatt Yolanda Rowe

Introduction

The *PlexStim™ Electrical Stimulator DLL Guide* provides a mechanism for user written programs to operate PlexStim Stimulator hardware. This document describes the functions and changes that have been made in the PlexStim.dll from revision 2.2.0.0 to revision 2.3.17.0. There are two very important notes as follows:

> **NOTE:** The revision 2.3.17.0 DLL contains several changes that will NOT be compatible with programs written for earlier versions of the DLL. User programs may need to be modified before they can function with the revision 2.3.17.0 DLL.
>
> As an example, the return value of ALL functions in the DLL is now an integer value that represents the error status of that function call. Furthermore all functions now report errors use a unified error reporting code (defined in PlexStimTypes.h). In previous versions of the DLL some functions returned nothing (void), an error code, a parameter value, or reported a combination of error code and parameter value. Changing the function return was necessary to incorporate error checking on the parameters passed to the functions. Many of the functions that have Major Changes are not essential to the operation of the stimulator and this reduces the potential impact of the changes.
>
> **NOTE:** The revision 2.3.17.0 DLL will not recognize stimulators manufactured prior to April 2015 that have not been upgraded. See the Hardware Compatibility section for additional details.

Hardware Compatibility

PlexStim.dll version 2.3.17.0 is designed to operate with PlexStim stimulators running firmware revision A (Firmware part number 14-20-A-07-A). Stimulators that were manufactured with revision blank firmware (14-20-A-07) will generally not be recognized by the 2.3.17.0 DLL as valid stimulators.

Stimulators of hardware revision F (14-20-A-10-F) are the first stimulators originally manufactured with revision A firmware. Examine the labels on the bottom of the stimulator to determine the firmware and hardware revision. If your stimulator is not revision F or later or does not have revision A or later firmware, contact Plexon support at support@plexon.com for information on upgrading the firmware in your stimulator.

> **NOTE:** After the stimulator Hardware/Firmware has been upgraded you will be required to use the revision 2.3.17.0 or later DLL to operate the stimulator as prior revisions of the DLL will not function properly with the upgraded hardware/firmware.

New Features

The 2.3.17.0 DLL now supports operation of up to four stimulators attached to the host computer simultaneously. The previous release of the PlexStim DLL, version 2.2.0.0, only supported the use of two stimulators attached to the computer.

Function Status

The following table lists all of the functions in the 2.3.17.0 library and indicates their status. Functions are listed in alphabetical order. The status indicates whether the function is new (New), has been changed in a major way (Major Changes), has been changed in a minor way (More Returns or New Return), or was not changed at all (Unchanged). Major Changes is a change in the way arguments are passed to or from the function. Existing programs that call the function with Major Changes will generally need to be modified. A minor change is an increase in the number of possible error codes returned by the function and/or a change in the meaning of the error codes. If the function returned an error code but now reports additional error codes, the status is More Returns. If the function returned nothing (void) but now reports an error code, the status is New Return. Existing programs can be modified to take advantage of the enhanced error reporting, but may not need to be modified.

Status	Function	Type
More Returns	int PS_Abort(int StimN)	Stimulation
New Return	int PS_AbortAll(void)	Stimulation
Major Changes	int PS_ChannelStimStarted(int StimN, int ChN, bool *bStarted)	Stimulation
New Return	int PS_CloseAllStim(void)	Initialization/Clean-up
More Returns	int PS_CloseStim(int StimN)	Initialization/Clean-up
More Returns	int PS_GetArbPatternPoints(int StimN, int ChN, int NPoints, int* Coords)	Pattern
More Returns	int PS_GetArbPatternPointsX(int StimN, int ChN, int NPoints, int* XCoords)	Pattern
More Returns	int PS_GetArbPatternPointsY(int StimN, int ChN, int NPoints, int* YCoords)	Pattern
New	iint PS_GetAutoDischarge(int StimN, bool *bDischarge)	Settings
New	int PS_GetDescription(int StimN, char *description)	Information
Major Changes	int PS_GetDigitalOutputMode(int StimN, PS_DIGITAL_OUTPUT *mode)	Settings
New	int PS_GetExtendedErrorInfo(int ErrorCode, char *ErrorString)	Information
New	int PS_GetFwVersion(int StimN, int *fw_version)	Information
Major Changes	int PS_GetMonitorChannel(int StimN, int *MonChN)	Settings
Major Changes	int PS_GetNChannels(int StimN, int *ChN)	Initialization/Clean-up
Major Changes	int PS_GetNPointsArbPattern(int StimN, int ChN, int *points)	Pattern
Major Changes	int PS_GetNStim(int *NumStim)	Initialization/Clean-up
Major Changes	int PS_GetPatternType(int StimN, int ChN, PS_PATTERN_TYPE *type)	Pattern
Major Changes	int PS_GetPeriod(int StimN, int ChN, double *period)	Settings
Major Changes	int PS_GetRate(int StimN, int ChN, double *rate)	Settings
Major Changes	int PS_GetRectParam(int StimN, int ChN, PS_RectPattern *pattern)	Pattern
Major Changes	int PS_GetRepetitions(int StimN, int ChN, int *R)	Settings
New	int PS_GetSerialNumber(int StimN, char *serial)	Information
Major Changes	int PS_GetStimPatternDuration(int StimN, int ChN, double *duration)	Settings
Major Changes	int PS_GetTriggerMode(int StimN, PS_TRIG_MODE *mode)	Settings
Major Changes	int PS_GetVmonScaling(int StimN, PS_VMON_SCALING *vmon_scaling)	Settings
Unchanged	int PS_InitAllStim(void)	Initialization/Clean-up
Major Changes	int PS_IsWaveformBalanced(int StimN, int ChN, bool *balanced)	Settings

Status	Function	Type
More Returns	int PS_LoadAllChannels(int StimN)	Loading Channel
More Returns	int PS_LoadArbPattern(int StimN, int ChN, char* s)	Pattern
More Returns	int PS_LoadChannel(int StimN, int ChN)	Loading Channel
New	int PS_SetAutoDischarge(int StimN, bool Enabled)	Settings
New Return	int PS_SetDigitalOutputMode(int StimN, PS_DIGITAL_OUTPUT mode)	Settings
More Returns	int PS_SetMonitorChannel(int StimN, int ChN)	Settings
New Return	int PS_SetPatternType(int StimN, int ChN, PS_PATTERN_TYPE type)	Pattern
New Return	int PS_SetPeriod(int StimN, int ChN, double period)	Settings
New Return	int PS_SetRate(int StimN, int ChN, double rate)	Settings
More Returns	int PS_SetRectParam(int StimN, int ChN, PS_RectPattern Param)	Pattern
New Return	int PS_SetRepetitions(int StimN, int ChN, int R)	Settings
New Return	int PS_SetTriggerMode(int StimN, PS_TRIG_MODE mode)	Settings
More Returns	int PS_SetVmonScaling(int StimN, PS_VMON_SCALING sc)	Settings
More Returns	int PS_StartStimAllChannels(int StimN)	Stimulation
More Returns	int PS_StartStimChannel(int StimN, int ChN)	Stimulation
More Returns	int PS_StopStimAllChannels(int StimN)	Stimulation
More Returns	int PS_StopStimChannel(int StimN, int ChN)	Stimulation

Functions with Major Changes

Several functions were changed such that the return value of the functions is now an error code. The former return value of the function is now an argument that is passed using a pointer. These changes will generally require a corresponding change in the user program that calls the function. Note, however, that most of the affected functions are “Get” functions that merely report back values that the user has previously set using the corresponding “Set” function. None of these functions communicate with the stimulator hardware, they only report the internal settings of the SDK. The use of these functions is thus somewhat optional. The functions PS_GetNStim and PS_IsWaveformBalanced are the exceptions. PS_GetNStim reports the number of stimulators that were connected to the computer, turned on, and of the correct revision (at the time PS_InitAllStim was called). PS_IsWaveformBalanced double checks the requested stimulation pattern to see if it is charge balanced. Use of this function is highly recommended to avoid accidental stimulation with waveforms that are not charge balanced.

Functions by Type

Information Functions

PS_GetExtendedErrorInfo - **NEW**

Current	int PS_GetExtendedErrorInfo(int ErrorCode, char *ErrorString)

This function reads an English language description of the ErrorCode that can be returned by all functions. See PlexStimTypes.h for the list of error codes that are in use.
You must provide a pointer to a string of at least 128 characters in length (char ErrorString[128]).
The return value of the function is an error code.

PS_GetDescription - **NEW**

Current	int PS_GetDescription(int StimN, char *description)

Reads the description or hardware version of the selected stimulator.
You must provide a pointer to a string of at least 64 characters in length (char description[64]).
The return value of the function is an error code.

PS_GetFwVersion - **NEW**

Current	int PS_ GetFwVersion(int StimN, int *fw_version)

This function queries the hardware and gets an integer representing the version of the firmware in the device as shown in the table below:

Firmware Version	Description
6	Revision A firmware (14-20-A-07-A) This firmware is compatible with PlexStim DLL 2.3.17.0
5	Revision blank firmware (14-20-A-07) This firmware is NOT compatible with PlexStim DLL 2.3.17.0 Contact Plexon support for a firmware upgrade

The return value of the function is an error code.

PS_GetSerialNumber - **NEW**

Current	int PS_GetSerialNumber(int StimN, char *serial)

Reads the serial number of the selected stimulator. When multiple stimulators are connected to the host computer, they are sorted such that the lowest serial number is stimulator #1.
You must provide a pointer to a string of at least 16 characters in length (char serial[16]).
The return value of the function is an error code.

Initialization/Clean-up Functions

PS_CloseAllStim - MORE RETURNS

Current	int PS_CloseAllStim(void)

Finalizes work with all available stimulators. Any stimulation in progress is aborted.
The return value of the funtion is now an error code.

PS_CloseStim - MORE RETURNS

Current	int PS_CloseStim(int StimN)

Finalizes work with stimulator StimN. Any stimulation in progress is aborted.
The return value of the funtion is now an error code.

PS_GetNChannels - **MAJOR CHANGES**

Current	int PS_GetNChannels(int StimN, int *ChN)
Former	int PS_GetNChannels(int StimN)

Gets maximum number of channels for stimulator.
The number of channels parameter (ChN) is now passed using a pointer.
The return value of the function is now an error code.

PS_GetNStim - **MAJOR CHANGES**

Current	int PS_GetNStim(int *NumStim)
Former	int PS_GetNStim(void)

Gets number of available stimulators. The maximum number of stimulators you can work with is four.
The number of stimulators parameter (NumStim) is now passed using a pointer.
The return value of the function is now an error code.

PS_InitAllStim - UNCHANGED

Current	int PS_InitAllStim(void)

Initializes all available stimulators and places them in stimulation mode (versus Z test mode).

Loading Channel Functions

PS_LoadAllChannels - MORE RETURNS

Current	int PS_LoadAllChannels(int StimN)

Loads parameters of all channels to the stimulator hardware.
The return value of the function is now an error code.

PS_LoadChannel - MORE RETURNS

Current	int PS_LoadChannel(int StimN, int ChN)

Loads parameters of channel ChN to the stimulator hardware.
The return value of the function is now an error code.

Pattern Functions

PS_GetArbPatternPoints – MORE RETURNS

Current	int PS_GetArbPatternPoints(int StimN, int ChN, int NPoints, int* Coords)

Gets X and Y coordinates of a graphical representation of the arbitrary waveform pattern loaded into the selected stimulator and channel. These coordinates can be used to draw a graph of the pattern.|
The return value of the function is now an error code.

PS_GetArbPatternPointsX – MORE RETURNS

Current	int PS_GetArbPatternPointsX(int StimN, int ChN, int NPoints, int* XCoords)

Gets X coordinates of a graphical representation of the arbitrary waveform contained in the loaded pattern file.
The return value of the function is now an error code.

PS_GetArbPatternPointsY – MORE RETURNS

Current	int PS_GetArbPatternPointsY(int StimN, int ChN, int NPoints, int* YCoords)

Gets Y coordinates of a graphical representation of the arbitrary waveform contained in the loaded pattern file.
The return value of the function is now an error code.

PS_GetNPointsArbPattern – **MAJOR CHANGES**

Current	int PS_GetNPointsArbPattern(int StimN, int ChN, int *points)
Former	int PS_GetNPointsArbPattern(int StimN, int ChN)

Gets the number of points in a graphical representation of the arbitrary waveform.
The pattern type parameter (type) is now passed using a pointer.
The return value of the function is now an error code.

PS_GetPatternType – **MAJOR CHANGES**

Current	int PS_GetPatternType(int StimN, int ChN, PS_PATTERN_TYPE *type)
Former	PS_RectPattern PS_GetRectParam(int StimN, int ChN)

Checks if channel set up to use rectangular pulse parameters or preloaded arbitrary waveform pattern.
The pattern type parameter (type) is now passed using a pointer.
The return value of the function is now an error code.

PS_GetRectParam – **MAJOR CHANGES**

Current	int PS_GetRectParam(int StimN, int ChN, PS_RectPattern *pattern)
Former	PS_RectPattern PS_GetRectParam(int StimN, int ChN)

Gets parameters of the rectangular pulse for a channel ChN.
The pattern parameter (pattern) is now passed using a pointer.
The return value of the function is now an error code.

PS_LoadArbPattern – MORE RETURNS

Current	int PS_LoadArbPattern(int StimN, int ChN, char* s)

Loads an arbitrary waveform pattern from a (.pat) file into a selected channel.
The return value of the function is now an error code.

PS_SetPatternType – NEW RETURN

Current	int PS_SetPatternType(int StimN, int ChN, PS_PATTERN_TYPE type)

Configures channel ChN to use rectangular pulse parameters (pattern = 0) or preloaded arbitrary waveform pattern (pattern = 1).
The return value of the function is now an error code.

PS_SetRectParam – MORE RETURNS

Current	int PS_SetRectParam(int StimN, int ChN, PS_RectPattern Param)

Sets parameters of the rectangular pulse for a channel.
The return value of the function is now an error code.

Settings Functions

PS_GetAutoDischarge - **NEW**

Current	int PS_GetAutoDischarge(int StimN, bool *bDischarge)

Reads the status of the automatic discharge function of the stimulator.
Refer to the *PlexStim™ Electrical Stimulator User Guide* and the AStAR™ System Guide (04-14-C-302) for additional details.
The return value of the function is an error code.

PS_GetDigitalOutputMode – **MAJOR CHANGES**

Current	int PS_GetDigitalOutputMode(int StimN, PS_DIGITAL_OUTPUT *mode)
Former	int PS_GetDigitalOutputMode(int StimN)

Checks if the Digital Output is low or high during the inter-pulse interval. Each stimulator channel has a dedicated digital output that indicates when stimulation is occurring on that channel. The digital output is always high during the pulse or arbitrary waveform output, but the user can control the state of the digital output during the time in between pulses or arbitrary waveforms.
The mode parameter (mode) is now passed using a pointer.
The return value of the function is now an error code.

PS_GetMonitorChannel – **MAJOR CHANGES**

Current	int PS_GetMonitorChannel(int StimN, int *MonChN)
Former	int PS_GetMonitorChannel(int StimN)

Gets number of channel set for display on the voltage and current monitor outplaces for stimulator StimN.
The monitor channel parameter (MonChN) is now passed using a pointer.
The return value of the function is now an error code.

PS_GetPeriod – **MAJOR CHANGES**

Current	int PS_GetPeriod(int StimN, int ChN, double *period)
Former	double PS_GetPeriod(int StimN, int ChN)

Gets period (milliseconds) for channel ChN.
The period parameter (period) is now passed using a pointer.
The return value of the function is now an error code.

PS_GetRate – **MAJOR CHANGES**

Current	int PS_GetRate(int StimN, int ChN, double *rate)
Former	double PS_GetRate(int StimN, int ChN)

Gets repetition rate for a channel in Hertz.
The rate parameter (rate) is now passed using a pointer.
The return value of the function is now an error code.

PS_GetRepetitions – **MAJOR CHANGES**

Current	int PS_GetRepetitions(int StimN, int ChN, int *R)
Former	int PS_GetRepetitions(int StimN, int ChN)

Gets number of repetitions - the number of times that the bi-phasic pulse or the arbitrary waveform (loaded from a text file) is repeated for channel ChN.
The repetitions parameter (R) is now passed using a pointer.
The return value of the function is now an error code.

PS_GetStimPatternDuration – **MAJOR CHANGES**

Current	int PS_GetStimPatternDuration(int StimN, int ChN, double *duration)
Former	double PS_GetStimPatternDuration(int StimN, int ChN)

Gets duration of the whole stimulation pattern.
The duration parameter (duration) is now passed using a pointer.
The return value of the function is now an error code.

PS_GetTriggerMode – **MAJOR CHANGES**

Current	int PS_GetTriggerMode(int StimN, PS_TRIG_MODE *mode)
Former	PS_TRIG_MODE int PS_GetTriggerMode(int StimN)

Gets mode for a stimulator StimN to start stimulation.
The mode parameter (mode) is now passed using a pointer.
The return value of the function is now an error code.

PS_GetVmonScaling – **MAJOR CHANGES**

Current	int PS_GetVmonScaling(int StimN, PS_VMON_SCALING *vmon_scaling)
Former	PS_VMON_SCALING PS_GetVmonScaling (int StimN)

Gets the voltage monitor scaling for stimulation mode.
The scaling parameter (scaling) is now passed using a pointer.
The return value of the function is now an error code.

PS_IsWaveformBalanced – **MAJOR CHANGES**

Current	int PS_IsWaveformBalanced(int StimN, int ChN, bool *balanced)
Former	bool PS_IsWaveformBalanced(int StimN, int ChN)

Gets duration of the whole stimulation pattern.
The balanced parameter (balanced) is now passed using a pointer.
The return value of the function is now an error code.

PS_SetAutoDischarge - **NEW**

Current	int PS_SetAutoDischarge(int StimN, bool Enabled)

Turns the automatic discharge function of the stimulator on or off (default = on).

> **CAUTION:** Turning the automatic discharge mode off is ONLY recommended in very specific situations when the stimulator is being used in conjunction with the AStAR System (Automatic Stimulation Artifact Recovery System).

The return value of the function is an error code.
Refer to the *PlexStim™ Electrical Stimulator User Guide* and the AStAR System Guide (04-14-C-302) for additional details.

PS_SetDigitalOutputMode – NEW RETURN

Current	int PS_SetDigitalOutputMode(int StimN, PS_DIGITAL_OUTPUT mode)

Sets Digital Output mode to low or high during inter-pulse interval. Each stimulator channel has a dedicated digital output that indicates when stimulation is occurring on that channel. The digital output is always high during the pulse or arbitrary waveform output, but the user can control the state of the digital output during the time in between pulses or arbitrary waveforms by using this function. Default value is 1 (low).
The return value of the function is now an error code.

PS_SetMonitorChannel – MORE RETURNS

Current	int PS_SetMonitorChannel(int StimN, int ChN)

Selects one channel (ChN) for display on the voltage and current monitor connectors for the stimulator StimN.
The return value of the function is now an error code.

PS_SetPeriod – NEW RETURN

Current	int PS_SetPeriod(int StimN, int ChN, double period)

Sets period for a channel in milliseconds. Default value is five ms.
The return value of the function is now an error code.

PS_SetRate – NEW RETURN

Current	int PS_SetRate(int StimN, int ChN, double rate)

Sets repetition rate for a channel in Hertz. Default value is 200 Hz.
The return value of the function is now an error code.

PS_SetRepetitions – NEW RETURN

Current	int PS_SetRepetitions(int StimN, int ChN, int R)

Sets number of repetitions - the number of times that the bi-phasic pulse or the arbitrary waveform (loaded from a text file) is repeated for channel ChN. Default value is 1.
The return value of the function is now an error code.

PS_SetTriggerMode – NEW RETURN

Current	int PS_SetTriggerMode(int StimN, PS_TRIG_MODE mode)

Sets mode for a stimulator StimN to start stimulation.
The return value of the function is now an error code.

PS_SetVmonScaling– MORE RETURNS

Current	int PS_SetVmonScaling(int StimN, PS_VMON_SCALING sc)

Set the voltage monitor scaling for the voltage monitor output in units of V/V.
The return value of the function is now an error code.

Stimulation Functions

PS_Abort – MORE RETURNS

Current	int PS_Abort(int StimN)

Causes stimulation on stimulator StimN to cease immediately even if there is a pulse or arbitrary waveform in progress. This is in contrast to stopping stimulation by calling PS_StopStimChannel or PS_StopStimAllChannels.
The return value of the function is now an error code.

PS_AbortAll – NEW RETURN

Current	int PS_AbortAll()

Causes all stimulation for all available stimulators to cease immediately even if there is a pulse or arbitrary waveform in progress. This is in contrast to stopping stimulation by calling PS_StopStimChannel or PS_StopStimAllChannels.
The return value of the function is now an error code.

PS_ChannelStimStarted – **MAJOR CHANGES**

Current	int PS_ChannelStimStarted(int StimN, int ChN, bool *bStarted)
Former	bool PS_ChannelStimStarted(int StimN, int ChN)

Checks if stimulation is started for channel ChN.
The return value of the function is an error code.

PS_StartStimAllChannels – MORE RETURNS

Current	int PS_StartStimAllChannels(int StimN)

Starts stimulation for all channels for the stimulator StimN.
The return value of the function is an error code.

PS_StartStimChannel – MORE RETURNS

Current	int PS_StartStimChannel(int StimN, int ChN)

Starts stimulation for channel ChN with previously configured parameters.
The return value of the function is an error code.

PS_StopStimAllChannels – MORE RETURNS

Current	int PS_StopStimAllChannels(int StimN)

Stops stimulation for all channels for the stimulator StimN.
The return value of the function is an error code.

PS_StopStimChannel – MORE RETURNS

Current	int PS_StopStimChannel(intStimN, int ChN)

Stops stimulation for Channel ChN.

The return value of the function is an error code.

About Plexon Inc

Plexon is a pioneer and leading innovator of custom, high-performance data acquisition, behavior and analysis solutions specifically designed for scientific research. We collaborate with and supply thousands of customers including the most prestigious neuroscience laboratories around the globe driving new frontiers in areas including basic science, brain-machine interfaces (BMI), neurodegenerative diseases, addictive behaviors and neuroprosthetics. Plexon offers integrated solutions for *in vivo* neurophysiology, optogenetics, and behavioral research – backed by its industry-leading commitment to quality and customer support. For more information, please visit www.plexon.com.

Sales Support

For Sales Support, email info@plexon.com or call +1 (214) 369-4957.

Technical Support

If after reviewing this document, you would still like to access Plexon's Technical Support, we are available via several communication channels. You are invited to reach us through email, on the phone, or even over Skype utilizing instant messaging, voice, and/or video as follows:

EMAIL	PHONE	INSTANT MESSAGING, VOICE OR VIDEO VIA SKYPE
support@plexon.com	8:30 a.m. to 5:00 p.m. Central Time +1 (214) 369-4957	8:30 a.m. to 5:00 p.m. Central Time Skype name: plexonsupport *Skype is a free service. For more information on Skype or to download the application, go to www.skype.com.*