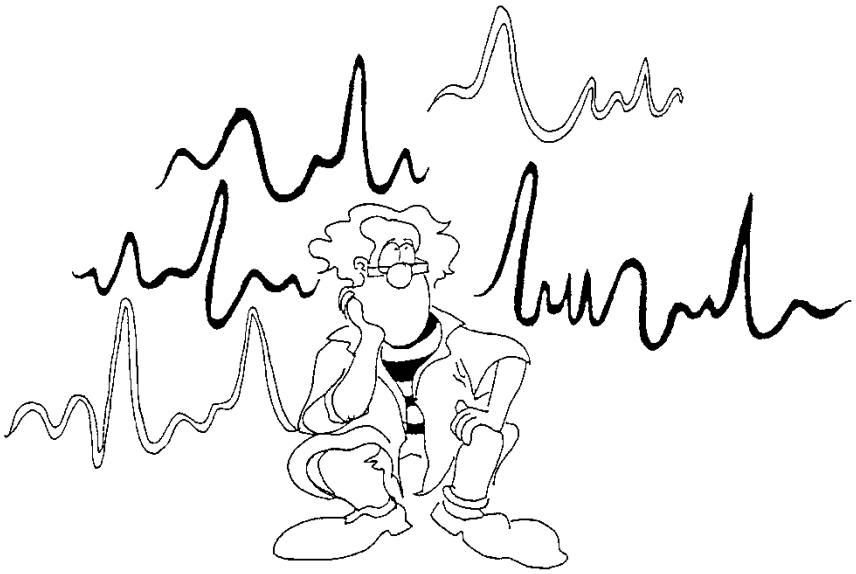
AlphaLab SnR/Neuro Omega

AO_MatlabTool

User's Manual

Version 1.2



ALPHA OMEGA
Home Office: P.O. Box 810, Nazareth Illit 17105, Israel
Email: info@alphaomega-eng.com
Web site: www.alphaomega-eng.com

Proprietary Notice

Contact Information:
ALPHA OMEGA ENGINEERING
P.O. Box 810
Nazareth Illit 17105
Israel
Tel: +972-4-656-3327
Fax: +972-4-657-4075
info@alphaomega-eng.com
support@alphaomega-eng.com
http://www.alphaomega-eng.com
U.S. Office: Toll Free: 1-877-919-6288, Fax: 1-877-471-2055
Europe Office: Toll Free: 00-800-2-574-2111, Tel +49-7-251-440-6620

Table of contents

1. Introduction

1.1. About This Manual

This manual describes the features and usage of the Matlab data access and control tool for AlphaLab SnR\Neuro Omega system.
This manual will use the AlphaLab SnR when describing the function, but all described functions also apply to Neuro Omega System.

1.2. Reference Documents

AlphaLab SnR Manual
Neuro Omega Manual

1.3. List of Abbreviations

SnR- Stimulation and Recording

1.4. Matlab Tool Overview

Matlab Tool is a set of commands allowing multi user communication with the AlphaLab SnR\Neuro Omega system. Each command is a Matlab function that allows the user to control certain system features.

Matlab Tool is compatible with Windows XP/7 32 bit.

1.5. Matlab Tool Command list

The table below shows a list of available Matlab function and a brief description of each one. The detailed usage of each command is explained later in the manual.

Function	***Description***
AO_StartConnection	Connect Matlab to AlphaLab SnR
AO_IsConnected	Check if Matlab is connected to the AlphaLab SnR
AO_CloseConnection	Close connection between Matlab and the AlphaLab SnR
AO_AddBufferingChannel	Acquire data for the specified channel
AO_GetAlignedData	Get aligned data from several channels

Function	***Description***
AO_GetChannelData	Get data for the specified channel
AO_ClearChannelData	Clear all buffered data
AO_GetNextBlock	Get next block of data
AO_SendBlock	Send stream format data
AO_StartSave	Start saving mpx file on the AlphaLab SnR
AO_SetSaveFileName	Set the mpx file name saved by AlphaLab SnR
AO_SetSavePath	Set the path of the directory where the lsx/mpx files are saved by AlphaLab SnR
AO_StopSave	Stop saving on the AlphaLab SnR
AO_SetStimulationParameters	Set stimulation parameters
AO_StartStimulation	Start stimulation
AO_StopStimulation	Stop stimulation
AO_SendDout	Send digital output to a specific Port in AlphaLab SnR

2. Getting started

This section describes the preparation work that needs to be done in order to be able to use the Matlab tool. Follow the steps below:

1. Install Matlab Tool

    a. Run the supplied setup file, and follow instructions

2. Start Matlab

    a. Open Matlab (Under Windows 7, you may need to run Matlab as Administrator, or change the user settings to lower Parental control)

    b. Set the working directory path in Matlab to the installed Matlab Tool Directory (Usually C:\Program Files\AlphaOmega\AO_MatlabTool)

3. Set up compiler and compile the mex file as follow:

    a. In Matlab command window type " mex –setup" and press enter; A Matlab message will appear in the command window:

    'Would you like mex to locate installed compilers [y]/n?]

    b. Press n
    The Matlab will suggest a list of all compilers it supports

    c. Choose a version of Microsoft visual such as Microsoft free visual c++ 2008 or 2010. *Note: If you do not have it on your PC, you need to install it before continuing (express mode is downloaded for free)*

    d. Continue procedure of choosing the compiler by answering the questions. This is for the path validation, so answer "y" to all questions if the path is correct.

4. Compile the Mex files as follows:

    a. In Matlab command window type "mex MexFileEthernetStandAlone.cpp Include\ethernetStandAlone.lib" and press enter; this command will compile the MexFileEthernetStandAlone.cpp and create a mex file called MexFileEthernetStandAlone.mexw32, and this completes the installation.

5. Test the installation as follow:

   a. In Matlab command window type: "AO_IsConnected" and press enter. If no compilation error, which is usually indicated by red colored messages, you are all set and ready to go.

      Otherwise,

   b. In Matlab command window type
      ! dependency walker\ depends.exe MexFileEthernetStandAlone.mexw32
      and press enter, and this should indicate which dll files are missing in order to run MexFileEthernetStandAlone.mexw32. If you are unable to obtain the missing dll's, contact Alpha Omega support.

Note:

In case of an old Matlab version, a compiling error will appear while compiling the cpp file. In this case:

a. Using any text editor, edit 'MexFileEthernetStandAlone.cpp' file

b. Go to the line #define NEW_MATLAB

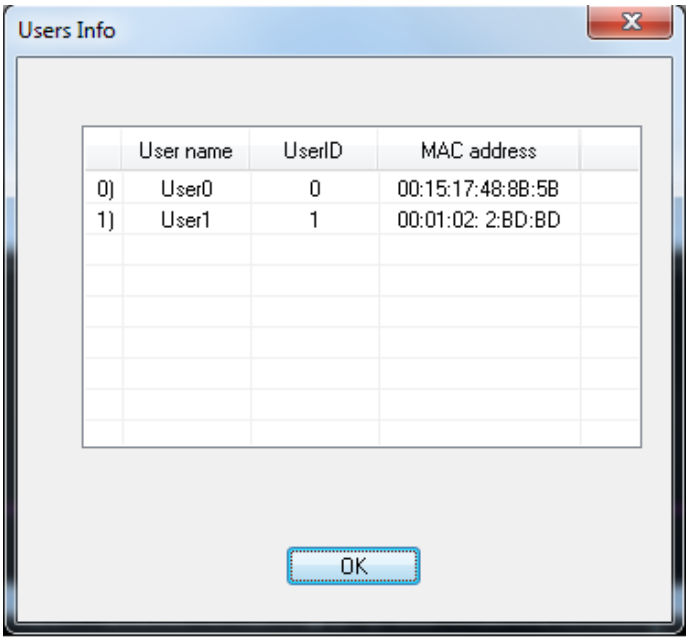
c. Add //at the beginning of the line to comment it out.(//#define NEW_MATLAB)

3. Matlab Functions and Usage

The table below gives complete syntax, description, and an example of each Matlab function.

Function	Function syntax and example
AO_StartConnection	**[Result] = AO_StartConnection(DspMac,PcMac,AdpaterIndex)** Used to connect Matlab to AlphaLab SnR system **Function parameters:** **DspMac**: String of 6 hex values. This is the mac address of the SnR system **PcMac**: String of 6 hex values. This is the mac address of matlab PC. This value must be different from the **DspMac** and the Mac address of the PC running the SnR software, it is recommended to find the actual mac address of the Matlab computer and use it. **AdpaterIndex**: Constant int value of -1 **Result**: Function return is an integer, 0 = no function errors, other number indicate function error (See appendix 1) It is preferable to ensure connection was done successfully by calling the function AO_IsConnected **%Example** PCMac='00:01:02:2:BD:BD'; DSPMac='00:21:ba:07:ab:9e'; Activeadapter=-1; retStartConnection=AO_startConnection(DSPMac,PCMac,Activeadapter); The following code should be added to insure proper connection:

Function	Function syntax and example

	for j=1:100,
	  pause(1);
	  ret=AO_IsConnected;
	  if ret==1
	  'The System is Connected'
	  break;
	  end
	end
	
	If connection was done successfully , the PcMac address will appear in AlphaLab SnR GUI => Help => User info
	
	 

AO_IsConnected	**[Results] = AO_IsConnected()** This function checks if Matlab is connected to the AlphaLab SnR Returns 1 if the system is connected, otherwise returns 0 **%Example** for j=1:100,

Function	Function syntax and example

	pause(1);
	ret=AO_IsConnected;
	if ret==1
	'The System is Connected'
	break;
	end
	end

AO_CloseConnection	**[Result] = AO_CloseConnection()**
	
	Function used to close connection between Matlab and the AlphaLab SnR system
	
	Result: Function returns an integer, 0 = no functionerrors, other number indicate function error (See appendix 1)
	
	% Example
	
	Result=AO_CloseConnection();
	if (Result==0)
	display('Connection closed successfully');
	else
	display('Connection close error');
	end

AO_AddBufferingChannel	**[Result] = AO_AddBufferingChannel(ChannelID,SamplingRate)** Function used to gather data for the channel defined in ChannelID **Function parameters:** **ChannelID**: The channel ID we want to gather data for **SamplingRate**: Sampling rate of the channel in samples/sec (this used strictly to calculate buffering information). **Result**: Function returns an integer, 0 = no function errors, other number indicate function error (See appendix 1)

Function	Function syntax and example
	Notes: • The size of buffering is 15sec* SamplingRate • The function stores the data using First In First Out (FIFO) mechanism • **The data value is A\D value including gain** _____ **%Example** ChannelID=10256; % set the channel number SamplingRate=44642.8571428; %set the sampling rate of the channel AO_AddBufferingChannel(ChannelID,SAmplingRate)% start gathering data for channel 10256 with sampling rate SamplingRate
AO_GetAlignedData	**[Result,pData,DataCapture,TS_FirstSample]= AO_GetAlignedData(ChannelIdArr,ChannelCount)** Function used to get aligned data for several channels. **Function parameters:** **ChannelIdArr**: Array of channels which we need to get data for, all channels must have the same sampling rate **ChannelCount**: Number of channels listed in **ChannelIdArr** **Result**: Function returns an integer, 0 = no function errors, other number indicate function error (See appendix 1) **pData**: Array of data samples of all listed channels, be arranged in single row **DataCapture**: The amount of the useable data captured in the **pData** array **TS_FirstSample**:The timestamp of the first sample for each channel **Notes:** • In order to get data you need to use **AO_AddBufferingChannel** first. • In order to get real time data u need to clear the buffered data first using **AO_ClearChannelData** function before using both commands

Function	Function syntax and example
	AO_GetAlignedData, **AO_GetChannelData** otherwise u will get stored data • **pData** will contain samples of data for all channels ,the number of valid samples in this array is **DataCapture** so make sure that you only get **DataCapture** samples. Hence, number of samples for each channel is **DataCapture** divided by **ChannelCount**. • In **pData** samples are arranged in a single array for all channels, starting with samples of the first channel listed in the **ChannelIdArr** , followed by other channels consecutively and in the same order. • **The data value is A\D including gain** **%Example** ChannelIdArr=[10000,10001,10002]; ChannelCount=3; [Result,pData,DataCapture,TS_FirstSample] = AO_GetAlignedData(ChannelIdArr,ChannelCount);% get aligned data from channels:10000,10001,10002 save them in the array pData, the alignment is done by time stamp TS_FirstSample
AO_GetChannelData	**[Result,pData,DataCapture] = AO_GetChannelData(ChannelId)** Function used to get data for the specified channel. **Function parameters:** **ChannelId:** The channel id which we want to get data for **pData:** Array of data **DataCapture:** The amount of the useable data in the array **Result:** Function returns an integer, 0 = no function errors, other number indicate function error (See appendix 1) **Notes:** • pData will contain a block of data

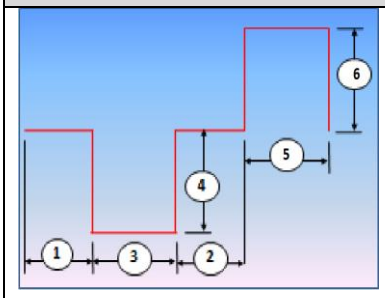
Function	Function syntax and example
	• The format of the data block is as follow: byte 1-2 : SizeOFtheBlock in words (1 word =2Byte) including the samples in this block byte 3 BlockType (in our case alwayes will be 'd' or 100) byte 4 Not used byte 5-6 ChannelNumber the id of the channel this block belongs to byte 7 Unit number ,this value valid only for segmented data byte 8 Not used byte 9-12 TimeStamp of the first sample of the block you will have to reorder them [byte10 byte9 byte12 byte11] byte 13-14 OverFlowCount the over flow of the time stamp – Future use byte 15-16 First sample byte 17-18 Second sample etc…. • In order to calculate the number of samples in this channel do the following HeaderSize=14bytes HeadrSizeWord=14bytes/2 **samplescount=SizeOFtheBlock-HeaderSizeWord = (SizeOFtheBlock-14)/2** **%Example** [Result,pData,DataCapture]=AO_GetChannelData(10256);
AO_ClearChannelData	**[Result] = AO_ClearChannelData()** Function used to clear buffered data by command **AO_AddBufferingChannel** **Result:** Function returns an integer, 0 = no function errors, other number indicate function error (See appendix 1)

Function	Function syntax and example
	Note: • In order to get real time data you need to clear the buffered data first using **AO_ClearChannelData** function before using both commands **AO_GetAlignedData , AO_GetChannelData** otherwise u will get stored data _____ **%Example** AO_ClearChannelData()
AO_GetNextBlock	**[Result,arraydata,realDataSizeWords]= AO_GetNextBlock(sizeOfArrayWords)** Function used to get the next new block data, the data should be parsed using **StreamFormat.h** file **Function parameters:** **sizeOfArrayWord**s: The max size of data the array can contain **arraydata:** Pointer to an array to hold the new data ,the data contain stream format in order to parse the data you need some Knowledge in our stream Format **realDataSizeWords**: The count of the data copied to the array data **Result:** Function returns an integer, 0 = no function errors, other number indicate function error (appendix 1) **Note:** • **StreamFormat.h** file is saved in the include directory _____ **%Example** realDataSizeWords=zeros(1,1); [res,arraydata,realDataSizeWords]=AO_GetNextBlock(50000);

Function	Function syntax and example
AO_SendBlock	**[Result] = AO_SendBlock(ArrayData)** Function used to send stream format data to the AlphaLab SnR system **Function parameters:** **ArrayDat:** Contain the data which will be sent to AlphaLab SnR system **Result**: Function return is an integer, 0 = no function errors, other number indicate function error (See appendix 1) **Notes:** • This function for advanced users only. • Stream format is explained in **StreamFormat.h** file **%Example** ArrayData=[7,1,2,3,4,5,6]; AO_SendBlock(ArrayData)
AO_StartSave	**[Result] = AO_StartSave();** Function used to start saving mpx file by the AlphaLab SnR system **Function parameters:** **Result:** Function returns an integer, 0 = no function errors, other number indicate function error (appendix 1) **Notes:** • The mpx file saved will contain the channels listed on the Data logging Options in the SnR • The filename and the saving path could be set before saving using Matlab commands: AO_SetSaveFileName, and AO_SetSavePath.

Function	Function syntax and example
	• Or by the parameters defined in the data logging(default) • When saving start the saving button in the Alpha Lab SnR GUI turns to red. See AlphaLab SnR Manual _____ **%Example:** [Result] = AO_StartSave()% start saving on the SnR/Neuro Omeg
AO_SetSaveFileName	**[Result] = AO_SetSaveFileName(FileName)** Function used to set mpx file name saved by AlphaLab SnR system **Function parameters:** **FileName:** Contains the file name. File name must be less than 30 chars **Result:** Function returns an integer, 0 = no function errors, other number indicate function error (See appendix1) _____ **%Example:** fileName='TestFile'; AO_SetSaveFileName(fileName)%set the file name as TestFile AO_StartSave()% start saving, the file name will be testFile
AO_SetSavePath	**[Result] = AO_SetSavePath(Path)** Function used to set the path of the directory to save in the mpx file **Function parameters:** **Path**: Contain the path of the directory for saving the files **Result**: Function return is an integer, 0 = no function errors, other number indicate function error

Function	Function syntax and example
	(See appendix1) _____ **%Example** path='c:\logging_data\' ;%the path of the directory to save in AO_SetSavePath(path)%set the path of the saving to 'c:\logging_data\' AO_StartSave()%start saving, the file will be saved at 'c:\logging_data\'
AO_StopSave	**[Result] = AO_StopSave()** Function used to stop saving by AlphaLab Snr/Neuro Omega system **Result:** Function returns an integer, 0 = no function errors, other number indicate function error (See appendix1) _____ **%Example** fileName='TestFile'; path='c:\logging_data\'; ;% AO_SetSaveFileName(fileName); AO_SetSavePath(path); AO_StartSave();% pause(100) ; AO_StopSave(); ➔saving will be done for 100 sec, the mpx file name is TestFile in the 'c:\logging_data\'
AO_SetStimulationParameters	**[Result] = AO_SetStimulationParameters (FirstPhaseDelay_mS,FirstPhaseAmpl_mA,FirstPhaseWidth_mS, SecondPhaseDelay_mS,SecondPhaseAmpl_mA,SecondPhaseWidth_mS,Freq_hZ,Duration_sec,ReturnChannel,channelnumber);** Function used to set stimulation parameters

Function	Function syntax and example
	**Function parameters:** **FirstPhaseDelay_mS:** First phase delay in mSec (1) **FirstPhaseAmpl_mA:** First phase amplitude (4) **FirstPhaseWidth_mS :** The width of the first phase (3) **SecondPhaseDelay_mS:** Second phase delay in mSec (2) **SecondPhaseAmpl_mA:** Second phase amplitude (6) **SecondPhaseWidth_mS:** Second phase width (5) **Freq_hZ:** The stimulation frequency **Duration_sec:** Duration of the stimulation after which stimulation stops **ReturnChannel:** The ID of the channel we want to return the stimulation with(set -1 for Global return) **channelnumber:** The channel we want to start stimulation on **Resuls:** Function returns an integer, 0 = no function errors, other number indicate function error (See appendix1) **Note:** • This function should be called before starting stimulation, otherwise stimulation will be done using the parameters defined in the SW GUI **%Example** FirstPhaseDelay_mS=1.1;%the delay of the first phase FirstPhaseAmpl_mA=-3.5;%the amp of the first phase FirstPhaseWidth_mS=0.5;%the width of the first phase SecondPhaseDelay_mS=1.5;%the delay of the second phase SecondPhaseAmpl_mA=1.5;%the amp of the second phase SecondPhaseWidth_mS=0.2;%the width of the second phase Freq_hZ=10;%the frequency of the stimulation Duration_sec=30;%duration of the stimulation ReturnChannel=10001;%the ID of the channel we want to return the stimulation with channelnumber=10000;%the channel we want to start stimualtion in AO_SetStimulationParameters(FirstPhaseDelay_mS,FirstPhaseAmpl_

Function	Function syntax and example
	mA,FirstPhaseWidth_mS,SecondPhaseDelay_mS,SecondPhaseAmpl_mA,SecondPhaseWidth_mS,Freq_hZ,Duration_sec,ReturnChannel,channelnumber);%set stimulation params
AO_StartStimulation	**[Result] = AO_StartStimulation(ChannelNumber);** Function used to start stimulation using AlphaLab SnR system **Function parameters:** **ChannelNumber:** The ID of the channel used for stimulation **Result:** Function returns an integer, 0 = no function errors, other number indicate function error (See appendix1) **Notes:** • The stimulation parameters should be set before stimulation using Matlab command **AO_SetStimulationParameters**, otherwise it will use the SW UI stimulation parameters • Be aware when stimulation is done with more than one channel, that the set stimulation parameters refer to the stimulation source • When stimulation is on, stimulation button on the GUI turn to red **%Example** ChannelNumber=10000; AO_StartStimulation(ChannelNumber);
AO_StopStimulation	**[Results]=AO_StopStimulation(ChannelNumber);** Function used to stops stimulation to the stimulation source of the **ChannelNumber** **Function parameters:**

Function	Function syntax and example
	ChannelNumber: The ID of the channel used for stimulation **Results:** Function returns an integer, 0 = no function errors, other number indicate function error (See appendix1) **Note:** In order to stop stimulation in all channels use: **ChannelNumber**= -1 _____ **%Example** ChannelNumber=10000; AO_StartStimulation(ChannelNumber);
AO_SendDout	**[Results] = AO_SendDout(DigtalChannelNumber,mask,Value)** Function used to send Dout to a specific Port **Function parameters:** **DigtalChannelNumber:** The ID of the digital port **Mask:** Is an 8 bit hex number input as a string. It indicates which bits are to be changed. Any bit set to 1 will be changed to the corresponding bit in the value, and any bit set to 0 will not be changed. **Value:** Can be any number between 0 and 2^8-1. **Results:** Function returns an integer, 0 = no function errors, other number indicate function error (See appendix1) _____ **%Example** DigtalChannelNumber=11701 ; %channel ID mask='0xFF'; %the mask (11111111) value=0; %the value (00000000) Result = AO_SendDOut(DigtalChannelNumber,mask,value); %Initialize all bits of port 11701 to 0

Function	Function syntax and example
	mask='0x05'; %the mask (00000101) value=3; %the value (00000011) Result = AO_SendDOut(DigtalChannelNumber,mask,value); %set port 11701 %====> The output of the bits on port 11701 will be '0000 0001' Mask=00000101 Value=00000011 Port= 00000001

4. Appendix-1 - Matlab function return cases

Function return	Result
0	No compiling error
1	The frequency must be bigger than 20
2	Duration of the signal is longer than 70 msec and its must be shorter
3	Null parameter
4	dll is not connected

5. Appendix-2: AlphaLab SnR\Neuro Omega channel list

Base addresses for each channel is defined in the workspace for example it could start with 10000 for first system, 20000 for second system, 30000 for third system and so on. Therefore, when addressing a specific channel you should address 10000+ channel number.

Number				
From	To	Signal	From / To	Type
Analog Inputs - MER, EEG, EMG….				
0	15	LFP, EEG (Cont) of Electrode 1..16	Connector 1, HS 1 on FE Box	Analog Input
16	31	LFP, EEG (Cont) of Electrode 17..32	Connector 2, HS1 or HS2 on FE Box	Analog Input
32	47	LFP, EEG (Cont) of Electrode 33..48	Connector 3, HS1 or HS2 on FE Box	Analog Input
48	63	LFP, EEG (Cont) of Electrode 49..64	Connector 4, HS1 or HS2 on FE Box	Analog Input
64	79	LFP, EEG (Cont) of Electrode 65..80	Connector 5, HS1 or HS2 on FE Box	Analog Input
80	95	LFP, EEG (Cont) of Electrode 81..96	Connector 6, HS1 or HS2 on FE Box	Analog Input
96	111	LFP, EEG (Cont) of Electrode 97..112	Connector 7, HS1 or HS2 on FE Box	Analog Input
112	127	LFP, EEG (Cont) of Electrode 113..128	Connector 8, HS1 or HS2 on FE Box	Analog Input
128	143	Segmented of Electrode 1..16	Connector 1, HS 1 on FE Box	Analog Input
144	159	Segmented of Electrode 17..32	Connector 2, HS1 or HS2 on FE Box	Analog Input
160	175	Segmented of Electrode 33..48	Connector 3, HS1 or HS2 on FE Box	Analog Input
176	191	Segmented of Electrode 49..64	Connector 4, HS1 or HS2 on FE Box	Analog Input
192	207	Segmented of Electrode 65..80	Connector 5, HS1 or HS2 on FE Box	Analog Input
208	223	Segmented of Electrode 81..96	Connector 6, HS1 or HS2 on FE Box	Analog Input
224	239	Segmented of Electrode 97..112	Connector 7, HS1 or HS2 on FE Box	Analog Input
240	255	Segmented of Electrode 113..128	Connector 8, HS1 or HS2 on FE Box	Analog Input
256	271	Spike, EMG (Cont) of Electrode 1..16	Connector 1, HS 1 on FE Box	Analog Input
272	287	Spike, EMG (Cont) of Electrode 17..32	Connector 2, HS1 or HS2 on FE Box	Analog Input
288	303	Spike, EMG (Cont) of Electrode 33..48	Connector 3, HS1 or HS2 on FE Box	Analog Input
304	319	Spike, EMG (Cont) of Electrode 49..64	Connector 4, HS1 or HS2 on FE Box	Analog Input
320	335	Spike, EMG (Cont) of Electrode 65..80	Connector 5, HS1 or HS2 on FE Box	Analog Input
336	351	Spike, EMG (Cont) of Electrode 81..96	Connector 6, HS1 or HS2 on FE Box	Analog Input
352	367	Spike, EMG (Cont) of Electrode 97..112	Connector 7, HS1 or HS2 on FE Box	Analog Input
368	383	Spike, EMG (Cont) of Electrode 113..128	Connector 8, HS1 or HS2 on FE Box	Analog Input
384	399	Raw (Cont) of Electrode 1..16	Connector 1, HS 1 on FE Box	Analog Input
400	415	Raw (Cont) of Electrode 17..32	Connector 2, HS1 or HS2 on FE Box	Analog Input
416	431	Raw (Cont) of Electrode 33..48	Connector 3, HS1 or HS2 on FE Box	Analog Input
432	447	Raw (Cont) of Electrode 49..64	Connector 4, HS1 or HS2 on FE Box	Analog Input
448	463	Raw (Cont) of Electrode 65..80	Connector 5, HS1 or HS2 on FE Box	Analog Input
464	479	Raw (Cont) of Electrode 81..96	Connector 6, HS1 or HS2 on FE Box	Analog Input

480	495	Raw (Cont) of Electrode 97..112	Connector 7, HS1 or HS2 on FE Box	Analog Input
496	511	Raw (Cont) of Electrode 113..128	Connector 8, HS1 or HS2 on FE Box	Analog Input
Analog Inputs - General Purpose				
1000	1015	General Purpose Analog Inputs 1..16	A. In connecor on FE Box	Analog Input
1016	1023	GP Analog Input 17..24	Analog Inputs Connectors (1-8) on Front Panel of RM Box and duplicated on A. In Connector (Chan 1-8) on Slot 1 on the Rear Panel of the RM Box	Analog Input
1024	1031	GP Analog Input 25..32	A. In Connecor (Chan 9-16) on Slot number 1 on the Rear Panel of the RM Box	Analog Input
User Defined Data Channels				
1032	1047	User Defined Analog Inputs	Internal - Future Use (Movement speed, channel calculation, other)	Analog Input
1050	1053	User Defined Waveform Data	Internal - can be sent to analog outputs or for stimlation	
1060	1075	User Defined Binary Data	Internal - Future Use	
Analog Outputs (Format will be different for scripting purposes)				
1100	1107	GP Analog Output 1-8	A. Out Connector on Front Panel of FE Box	Analog Output
1108	1115	GP Analog Output 9-16	Analog Outputs Connectors 1-8 on Front Panel of RM Box	Analog Output
1120	1121	Audio Output 1, right and left	Audio 1 on Front Panel of FE Box	Analog Output
1122	1123	Audio Output 2, right and left	Audio 2 on Front Panel of FE Box	Analog Output
1130	1131	Audio Output 1, right and left	Audio 1 on Front Panel of RM Box	Analog Output
1132	1133	Audio Output 2, right and left	Audio 2 on Front Panel of RM Box	Analog Output
Digital Inputs				
1200		Digital Port 1 with stobe and ready	D. In Port Connector on Front Panel of FE Box	Digital Input Port
1201		Digital Port 2	Internal on FE BOX - Not available as a port	
1202		Digital Port 3 with stobe and ready	D. In Port Connector on Rear Panel of RM Box - Slot 1	Digital Input Port
1203		Digital Port 4	Internal on RM BOX - Not available as a port	
1204		Digital Port 5 with stobe and ready	D. In Port Connector on Rear Panel of RM Box - Slot 2	Digital Input Port

1205		Digital Port 6	Internal on RM BOX - Not available	
1206		Digital Port 7 with stobe and ready	D. In Port Connector on Rear Panel of RM Box - Slot 3	Digital Input Port
1207		Digital Port 8	Internal on RM BOX - Not available	
1208		Digital Port 9 with stobe and ready	D. In Port Connector on Rear Panel of RM Box - Slot 4	Digital Input Port
1209		Digital Port10	Internal on RM BOX - Not available	
1210		Digital Port 11 with stobe and ready	D. In Port Connector on Rear Panel of RM Box - Slot 4	Digital Input Port
1211		Digital Port 12	Internal on RM BOX - Not available	
1212		Digital Port 13 with stobe and ready	D. In Port Connector on Rear Panel of RM Box - Slot 6	Digital Input Port
1213		Digital Port 14	Internal on RM BOX - Not available	
1214		Digital Port 15 with stobe and ready	D. In Port Connector on Rear Panel of RM Box - Slot 7	Digital Input Port
1215		Digital Port 16	Internal on RM BOX - Not available	
1216		Digital Port 17 with stobe and ready	D. In Port Connector on Rear Panel of RM Box - Slot 8	Digital Input Port
1217		Digital Port 18	Internal on RM BOX - Not available	
1218		Digital Port 19 with stobe and ready	D. In Port Connector on Rear Panel of RM Box - Slot 9	Digital Input Port
1219		Digital Port 20	Internal on RM BOX - Not available	
1220		Digital Port 21	Event Data	
1221		Digital Port 22	Stim Marker 1	
1222		Digital Port 23	Stim Marker 2	
1223		Digital Port 24	Stim Marker 3	
1224		Digital Port 24	Stim Marker 4	
1300	1315	TTL Input - Port1, bit 1-16	D.In Port Connector on Front Panel of FE Box	TTL Input
1316	1319	TTL Input - Port 2, bit 1-4	D.In Connector on Front Panel of FE Box	TTL Input
1320	1331	TTL Input - Port 2, bit 5-16	Not in use	
1332	1347	TTL Input - Port 3, bit 1-16	D.In Port Connector on Rear Panel of RM Box - Slot 1	TTL Input
1348	1351	TTL Input - Port 4, bit 1-4	Digital Inputs connectors 0-3 on Front Panel of RM Box	TTL Input
1352	1363	TTL Input - Port 4, bit 5-16	Not in use	
1364	1379	TTL Input - Port 5, bit 1-16	D.In Port Connector on Rear Panel of RM Box - Slot 2	TTL Input
1380	1395	TTL Input - Port 6, bit 1-16	Not in use	
1396	1411	TTL Input - Port 7, bit 1-16	D.In Port Connector on Rear Panel of RM Box - Slot 3	TTL Input
1412	1427	TTL Input - Port 8, bit 1-16	Not in use	
1428	1443	TTL Input - Port 9, bit 1-16	D.In Port Connector on Rear Panel of RM Box - Slot 4	TTL Input
1444	1459	TTL Input - Port 10, bit 1-16	Not in use	
1460	1475	TTL Input - Port 11, bit 1-16	D.In Port Connector on Rear Panel of RM Box - Slot 5	TTL Input
1476	1491	TTL Input - Port 12, bit 1-16	Not in use	
1492	1507	TTL Input - Port 13, bit 1-16	D.In Port Connector on Rear Panel of RM Box -	TTL Input

			Slot 6	
1508	1523	TTL Input - Port 14, bit 1-16	Not in use	
1524	1539	TTL Input - Port 15, bit 1-16	D.In Port Connector on Rear Panel of RM Box - Slot 7	TTL Input
1540	1555	TTL Input - Port 16, bit 1-16	Not in use	
1556	1571	TTL Input - Port 17, bit 1-16	D.In Port Connector on Rear Panel of RM Box - Slot 8	TTL Input
1572	1587	TTL Input - Port 18, bit 1-16	Not in use	
1588	1603	TTL Input - Port 19, bit 1-16	D.In Port Connector on Rear Panel of RM Box - Slot 9	TTL Input
1604	1619	TTL Input - Port 20, bit 1-16	Not in use	
Digital Outputs				
1700		GP Digital Output Port 1 (Bits 1-8)	D. Out Connector on Front Panel of FE Box	TTL Output
1701		GP Digital Output Port 2 (Bits 1-8)	Digital Outputs Connectors 1-8 on Front Panel of RM Box	TTL Output
1702		GP Digital Output Port 3 (Bits 1-8)	Internal on ADIO Board installed in Slot 2 of RM Box	TTL Output
1703		GP Digital Output Port 4 (Bits 1-8)	Internal on ADIO Board installed in Slot 3 of RM Box	TTL Output
1704		GP Digital Output Port 5 (Bits 1-8)	Internal on ADIO Board installed in Slot 4 of RM Box	TTL Output
1705		GP Digital Output Port 6 (Bits 1-8)	Internal on ADIO Board installed in Slot 5 of RM Box	TTL Output
1706		GP Digital Output Port 7 (Bits 1-8)	Internal on ADIO Board installed in Slot 6 of RM Box	TTL Output
1707		GP Digital Output Port 8 (Bits 1-8)	Internal on ADIO Board installed in Slot 7 of RM Box	TTL Output
1708		GP Digital Output Port 9 (Bits 1-8)	Internal on ADIO Board installed in Slot 8 of RM Box	TTL Output
1709		GP Digital Output Port 10 (Bits 1-8)	Internal on ADIO Board installed in Slot 9 of RM Box	TTL Output
1800		Flushing channel	Internal Use	
1999		Steam Data	Internal Use	