42 Aero Camino, Goleta, CA 93117
Tel (805) 685-0066 | Fax (805) 685-0067
info@biopac.com | **www.biopac.com**

06.14.13

Application Note 244: Programmable Stimulator for E-Prime®

This application note describes how to use BIOPAC stimulators (STMISOL/STMISOLA or STMISOC), E-Prime® experiment generation software, and a Measurement Computing Corporation (MCC) data acquisition device (USB-1208HS-4AO) to execute stimulus delivery. These tools allow one computer to seamlessly control audio and/or visual stimulus presentation, human subject feedback, and electrical stimulation. The stimulator produces pulses (or other waveforms) with heights and durations determined by E-Prime®. E-Prime® can establish stimulation times and amplitudes based on feedback during an experiment. The MCC device interfaces the computer and the stimulator. Experiments designed in E-Prime® can control up to four stimulators independently with one acquisition device.

Equipment needed:

This complete setup is available from BIOPAC as Programmable Stimulation System for E-Prime® - STMEPM

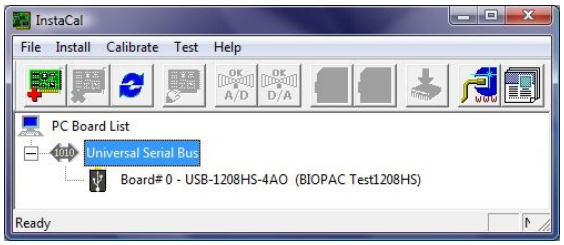
- Stimulator (STMISOL, STMISOLA, STMISOC)
  - Up to four stimulators can be used.
  - STMISOC also requires STM100C and either IPS100C or an MP system (MP100 or MP150)
- Data acquisition device
  - Software described in this document was tested with Measurement Computing Corporation (MCC) **USB-1208HS-4AO**. Other MCC devices may work with little or no modification to the software thanks to MCC's Universal Library.
- E-Prime® experiment generation software
  - Scripts were written using E-Prime® 2.0 and script text is included on the BIOPAC distribution CD. Earlier versions of E-Prime® can likely run the scripts (not tested)—see notes on page 8.
  - BIOPAC distributes E-Prime® for Psychology Software Tools, Inc.
- Interface cable(s): stimulator to MCC device

Initial Setup

The following steps must be performed once to enable the USB-1208HS-4AO. If, while running an E-Prime® program, you receive an error message indicating that the data acquisition hardware is not recognized, re-performing step 2 may resolve the issue.

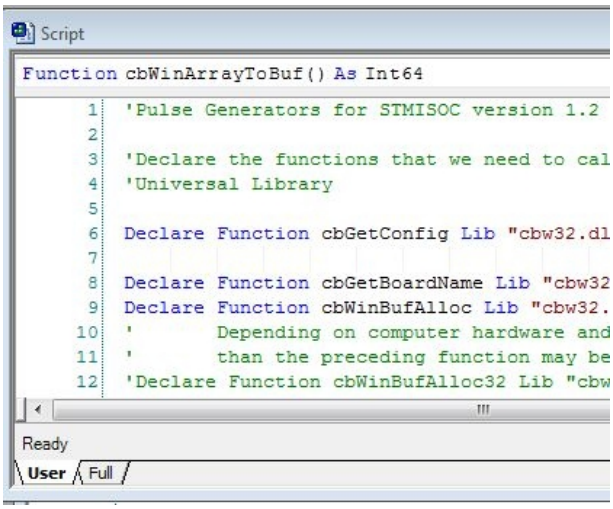
1. On the machine that will present stimuli, install the software.
   a. *Insta*Cal and the Universal Library must be installed from the MCC DAQ disk that came with the Measurement Computing hardware (it is also available at the Measurement Computing website).
      - Follow instructions in the Quick Start Guide that accompanied the MCC hardware.
   b. E-Prime® software – sold separately from STMEPM.
2. Run *Insta*Cal with the device connected to the computer.

a. Within *Insta*Cal, choose “Configure...” from the “Install” menu. This action generates a configuration file that is specific to the connected hardware and necessary for the operation of the Universal Library.



Implementation - Overview

The BIOPAC distribution CD contains four sample programs that can be used for testing or modified for use in your own experiments. Each of the sample programs has two versions, one for STMISOL/STMISOLA, and another for STMISOC. For your experimental purposes, the most important components of these programs are the definitions in the User section of the script. To find this code, open one of the sample programs in E-Studio and make sure that there is a check mark next to “Script” in the “View” menu. If there is not, you may select it in that menu or type Alt-5. With the script window open in E-Studio, click the “User” tab at the bottom left to expose the User script. A User.txt file is also included with each of the sample programs. This file contains the same code found in the User section of the script for that program.



The code in the User section of the script can be copied to make the functions defined within available to your own programs. Three of the programs, StimAndShow, USB1208HS_AOut and USB1208HS_AOutSimple all contain the same definitions in the User Script. They differ in the user interfaces that allow the defined functions to be called. You should run at least one of the sample programs and then look at the code that calls the functions to see how electrical stimulation can be programmed.

The functions in these three programs produce square pulses. As described in greater detail in later sections, one of the functions, SendAOutPulse, is designed to send a single pulse. That pulse can be directed to up to four stimulators simultaneously. The timing will be the same for all stimulators (if there is more than one), but the amplitudes can be adjusted independently.

Another function in these programs, SendAZeroPulses, can be used to send a train of pulses to one stimulator.

The fourth program, USB1208HS_AOutSine, contains the function SendAOutSine that produces sine waves instead of single pulses. Like USB1208HS_AOut, this function can send its output to as many as four stimulators at once.

These specific steps can be followed, as an example, to make the functions SendAOutPulse, SendAZeroPulses, and (for STMISOC only) SetStimulatorZero available to your own programs:

1. Use E-Studio to open the file **USB1208HS_AOut.es2** (and see the individual components).
   - This file provides interface commands to communicate with the D/A unit.
   - The most important part of the file, the part which may be incorporated into experiment files, is the text in the User part of the Script.
   - This text is also included in a file called **UserScript.txt** in the AOut folder.
2. Copy and paste this text into the User section of the script in your own program.
3. Call the function “SendAOutPulse” or “SendAZeroPulses” from InLine objects in the experiment file.

Implementation – Incorporating the Provided Functions

In this section, the basic functions in the sample files are described in greater detail. In most respects the functions are the same for both STMISOL/STMISOLA and STMISOC. They are housed in separate folders on the installation CD, however, because important differences in the operation of the two types of stimulator make it critical that functions written for one stimulator not be used for the other. Functions defined specifically for STMISOC have “STMISOC” added to the end of their names to make it easier to distinguish them. That part of the function name is suppressed in this document.

SendAOutPulse:

“SendAOutPulse” returns a value of type Long, and requires four inputs. The function causes the stimulator to deliver square pulses. The inputs specify the parameters of these pulses.

1. *First input:* The first input specifies the pulse duration. This input must be formatted as a double precision floating point number and will be interpreted as a time in milliseconds.

    - As a safety feature, the software checks to see if the duration is overly long. If a pulse duration longer than 1 msec is passed, the software displays a warning to confirm the stimulus length is actually desired. This feature is configurable through another statement in the User Script:

    "Const MaxDuration# = 1". To suppress the warning, increase the value specified by this constant. The STMISOC is not capable of producing pulses longer than 2 milliseconds, so for programs written for that stimulator, MaxDuration should never be larger than 2.

2. *Second input:* The second input to "SendAOutPulse" is a vector of integers indicating the channels through which the stimulus will be sent. The numbers correspond to the AOUT labels on the USB1208HS. For instance, to control stimulation through AOUT0 and AOUT2, pass the values 0 and 2 through the second input to "SendAOutPulse".

3. *Third input:* The third input can be a scalar or a vector of voltages. This input must be of type double precision.

4. Fourth input: The fourth input is an integer specifying the operating mode of the stimulator. STMISOL has two options ("V" and "I"), STMISOLA has three ("V" and two "I" settings distinguished by an additional toggle switch). STMISOC has three options ("Voltage (1:10)", "Voltage (1:5)", and "Current"). Possible modes are defined as constants in the User section of the E-Prime® script.

    - The software assumes that the values specified in the third argument are the voltages that the stimulator will produce. The STMISOL in "V" mode amplifies its input by a factor of 20, so the voltages output by the USB1208HS will be a factor of 20 below the voltages specified by this input when in that mode.

To send the same voltage through all of the channels listed in the second input, send a single number through the third input. Otherwise, the number of elements in the voltage vector must agree with the number of elements in the channel vector. As an example, both vectors can be of length two in order to send two different voltages through two channels of the USB1208HS. For instance, to send a 57 V pulse through channel 0 and a 48 V pulse through channel 2, part of the script in an InLine object might look like this:

```
Dim PulseDur As Double
Dim SelChans() As Integer
Dim Voltages() As Double
Dim ReturnValue As Long
Dim OutputMode As Integer

ReDim SelChans(1)
ReDim Voltages(1)

PulseDur = 1
SelChans(0) = 0
SelChans(1) = 2
Voltages(0) = 57
Voltages(1) = 48
OutputMode = Voltage

ReturnValue = SendAOutPulse(PulseDur, SelChans, Voltages, OutputMode)
```

As an added safety measure, the voltage is limited just like the pulse duration. To produce voltages larger than 85 volts without a warning, modify this line in the User Script:

```
Const MaxVoltage# = 85
```

SendAZeroPulses:

"SendAZeroPulses" returns a value of type Long and requires four inputs (five for STMISOC). The function causes a series of square pulses to be sent through channel AOUT0 of the USB1208HS. The four input arguments determine the parameters of this series of pulses.

1. *First Input:* The first argument specifies the duration of all pulses in milliseconds, just like "SendAOutPulse", and has the same built-in protection against overly long pulses.

2. *Second Input:* The number of pulses to produce in the stimulus train; must be specified as an integer.

3. *Third Input:* (*See Fifth input for an important note for STMISOC*) The output voltage is specified as a double precision scalar. All pulses in the train will have the same amplitude. This input has the same built-in protection against high voltages as does "SendAOutPulse". The amplitude is scaled in units of the output. For STMISOL in V mode, voltages produced by the USB1208HS are factor of 20 lower than the number specified by this input.
4. *Fourth Input:* The fourth input is the rate of the pulses. It must be a double precision number specifying the frequency in pulses per second. If the frequency is too high relative to the pulse duration, an error will be generated. The frequency must be low enough that a subsequent pulse does not start before or at the end of the previous pulse.
5. *Fifth Input:* The above sequence is correct for STMISOL. The function as written assumes that all STMISOL/STMISOLA's are operating in "V" mode. For STMISOC, the **third input argument** is an integer specifying which of the two voltage modes the stimulator is in. For STMISOC, the fourth input is the output voltage, and the fifth input is the pulse rate.

SendAOutSine:

"SendAOutSine" returns a value of type Long and requires four inputs (five for STMISOC). These inputs establish the characteristics of the temporary sine wave that the USB1208HS, and hence the stimulators it drives, will produce.

1. *First Input:* The first argument is a double precision floating point number specifying the duration of the sine wave in milliseconds.
2. *Second Input:* The second input is an array of integers specifying the channels that will output sine waves. The format is the same as that of the second input of "SendAOutPulse".
3. *Third Input:* The third input must be a double precision floating point number indicating the voltage at the peak of the sine wave. The voltage is specified in units of the output of the stimulator.
4. *Fourth Input:* (*See Fifth input for an important note about STMISOC*) The final input to SendAOutSine is a double precision floating point number specifying the frequency of the sine wave to be produced. It should be specified in Hz.
5. *Fifth Input*: For STMISOL/STMISOLA, the device is presumed to be operating in "V" mode. For STMISOC, the f**ourth input** is a vector of voltage multipliers (each either 5 or 10 depending upon the settings of the associated stimulators). The fifth input is the frequency of the sine wave.

SetStimulatorZero:

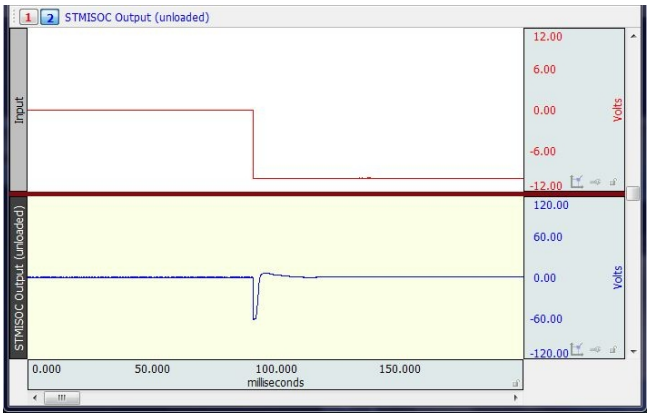
"SetStimulatorZero" – *for STMISOC **only*** – returns a value of type Long and requires two inputs. After this function has been run, the USB1208HS channels will be altered to new DC levels. The device will hold these voltages until another call to "SetStimulatorZero" or until the device is disconnected or the computer rebooted.

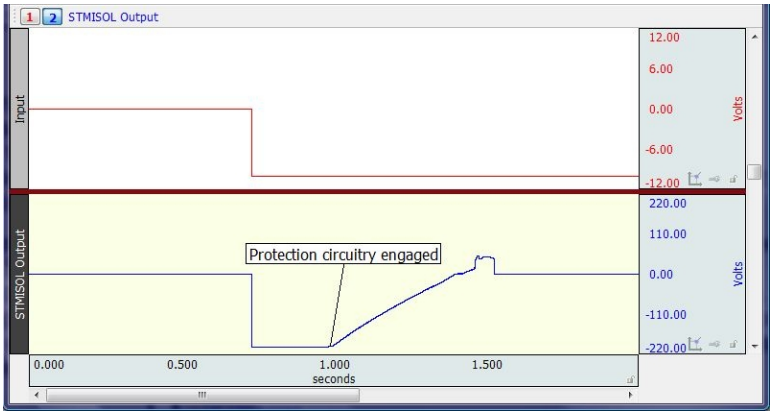
1. *First Input:* The first input is an array of channel numbers. The array should contain integers and uses the same specification as the second input to "SendAOutPulse".
2. *Second Input:* The second input should be a scalar or an array of the same length as the first input. It must be formatted as double precision floating point. This argument sets the new DC level(s) in units of volts supplied by the USB1208HS.

Implementation – Example User Interface

USB1208HS_AOut includes InLine script objects that produce a graphical interface to specify voltage and timing parameters for electrical stimulation. Please note that this is a sample program designed to help you test the equipment and ensure proper communication between E-Prime® and the MCC USB-1208HS-4AO. This program is **not** intended for use during your own experiments. However, it is intended to provide guidance that will help programmers understand how to construct software that will safely and effectively operate their stimulators.

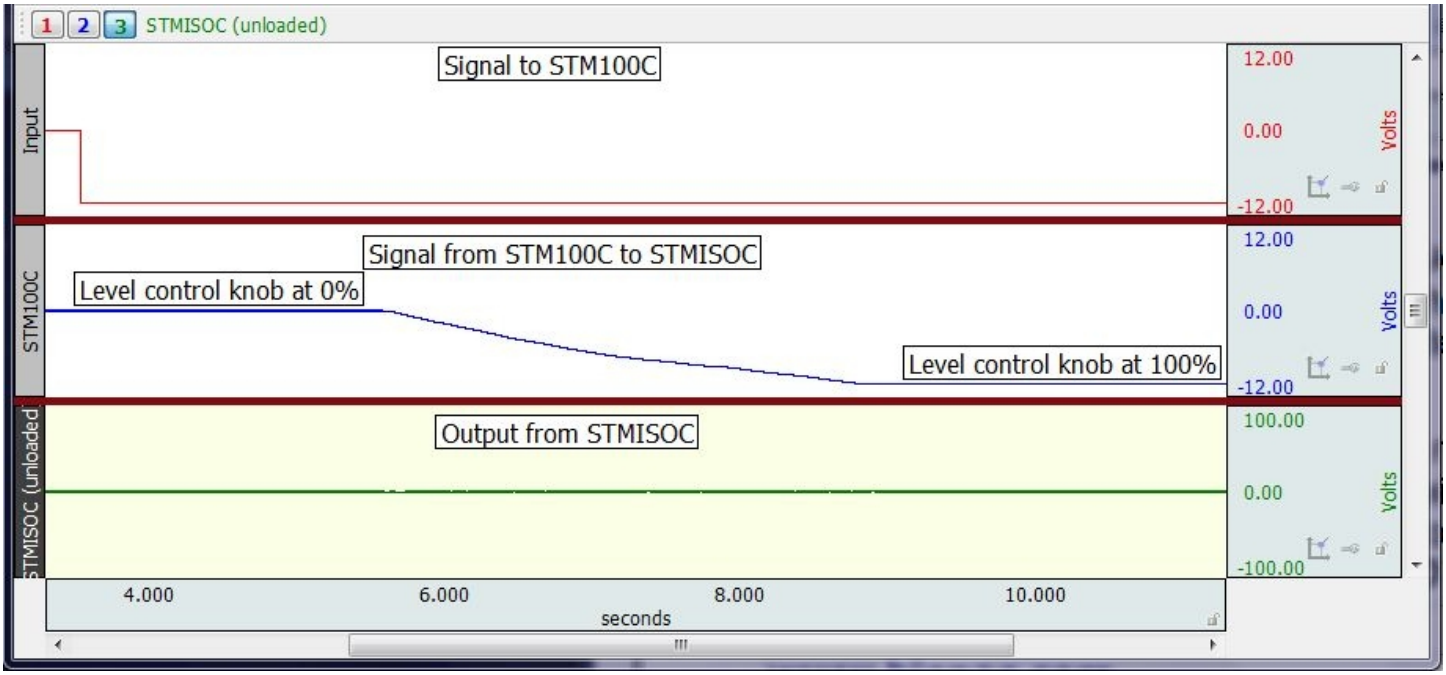
There is a major difference between the operating characteristics of the STMISOL/STMISOLA and the STMISOC. When driven by a non-zero DC potential, the STMISOL/STMISOLA will output an amplified copy of that DC potential. Consequently the input to a STMISOL or STMISOLA should be held at zero volts except during intentional stimulation. The STMISOC is AC-coupled. If the input to a STMISOC is suddenly changed to a new DC level, the device will respond with an amplified transient; the voltage will change suddenly but then settle back down to zero within a few milliseconds even if the input voltage is held steady. This difference is depicted in the images below.



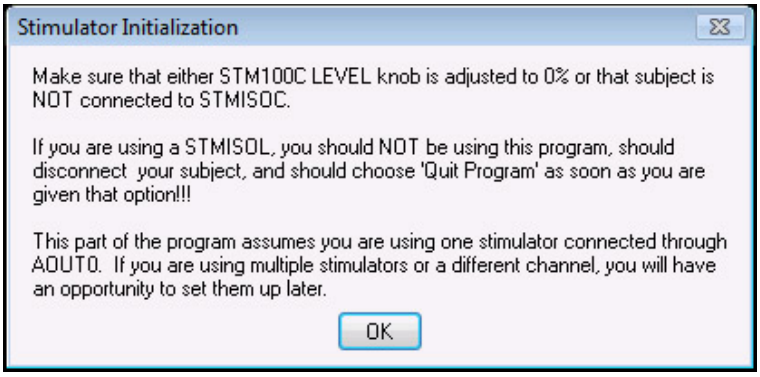


Both images show the output of the USB-1208HS-4AO (upper trace in red). The voltage begins at zero and then quickly drops to and holds -10 volts. The lower traces represent the outputs of the stimulators to which the USB1208HS is connected. The left image shows the output from a STMISOC; the right image shows that of a STMISOL. Note the drastically different time scales. Automatic protection circuitry eventually turns off the output from the STMISOL. However, before that happens the device holds -200 volts for approximately 250 milliseconds. The device will hold lower but still potentially painful voltages indefinitely. In contrast, the STMISOC reaches less than half the voltage and holds for less than 1/100th as long. Even this, however, can be painful to subjects, and software should be written to prevent accidentally stimulating participants during preparation of an experiment.

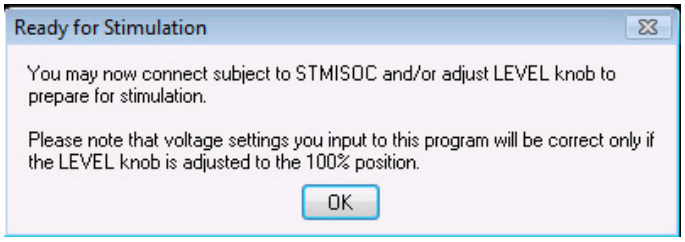
Under ordinary circumstances, one would not want to apply a DC voltage to the input of a STMISOL. On the other hand, the STMISOC may regularly be supplied with a DC offset because this is how to produce the largest possible range of output levels. Users may generally want to set the STMISOC input level to -10 volts at the start of any experiment. This level should be set when the subject is ***NOT*** connected to the electrodes, or when the LEVEL control knob on the STM100C is used to protect the subject. The following figure shows how adjustment of the LEVEL control can be used to prevent undesirable transients from reaching the subject. The voltage to the STM100C may be suddenly dropped to -10 volts with the LEVEL control set to pass 0% of this input. Then the LEVEL is smoothly adjusted to 100%. Because the voltage input to the STMISOC varies too slowly for the device to follow, the STMISOC never produces any stimulation that might cause a subject discomfort.



All sample programs intended for use with STMISOC except for USB1208HS_AOutSine begin by setting the input to the STM100C at -10 volts via a call to SetStimulatorZero. A graphical message (below left) is provided to the user to ensure that one of the two methods described above is utilized to shield the subject from the potential effects of this transition.

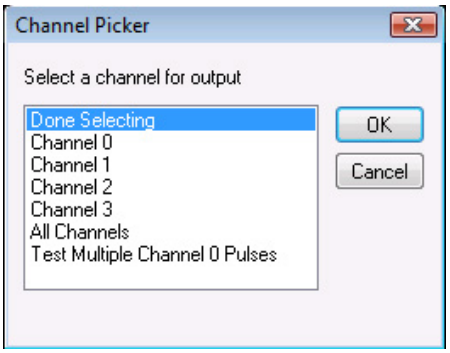


Upon clicking "OK", the user is presented with an additional informational dialog box as shown below.



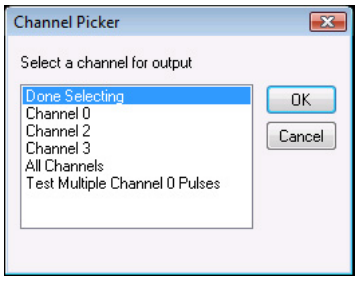
Note that hardware connections for the STMISOC and STMISOL/STMISOLA are different in that STMISOC is always driven by a STM100C. The LEVEL control knob on that device allows the user to diminish the input level to the STMISOC, so voltage levels are not fully under software control. The voltages presented to the USB-1208HS-4AO by E-Prime® in the example programs are nominal only for the case that the level control knob is turned to 100% during stimulation. It should also be noted that the STMISOC's actual voltage levels are dependent to some extent upon the resistive load across its outputs. It is recommended that the Voltage Monitor output of the STMISOC be recorded if knowledge of the actual output voltages is desired.

USB1208HS_AOut:

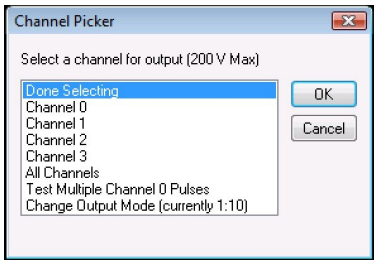


The user interface provided by running USB1208HS_AOut allows a fair amount of flexibility to the user. However, in general, there is an expected sequence of steps the user will perform to generate stimulation. The first step is the selection of a channel or channels through which stimulation will be provided. Channel selection proceeds through a dialog box as shown at right. This image was created with a USB-1208HS-4AO, so there are four channels available for selection.

- Only one item at a time can be selected. To test a subset of all channels, select a channel and click "OK" (or double-click the channel), and then select another channel when the dialog is regenerated.

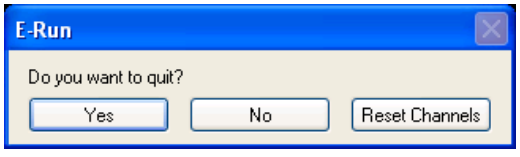


- Once a channel has been selected, it is no longer included as a choice. In the figure to the left, Channel 1 has been selected and so it does not appear as a possible selection.
- Select "All Channels" to select all of the analog output channels of the device and proceed to the next step.
- If not selecting "All Channels", then select "Done Selecting" after clicking "OK" for the last desired choice.
- "Test Multiple Channel 0 Pulses" calls "SendAZeroPulses" directly with parameters that can only be modified by editing the program (i.e., not via the user interface). All other selections lead toward a call to "SendAOutPulse".
- For STMISOC, there is an additional choice as indicated in the image to the right. The STMISOC has two voltage output modes, 1:10 and 1:5. To tell the software which position the mode switch is in, "Change Output Mode" is provided in this dialog box. Selecting that item toggles the mode between the two choices.



- Click "Cancel" to deselect a channel if a mistake was made. "Cancel" will generate a quit prompt as shown below left.
  - Click "Yes" to quit the program.
  - Click "No" to continue on toward stimulus delivery. This option is equivalent to the choice of "Done Selecting" from the channel selection dialog box.

Cancel generates this dialog:

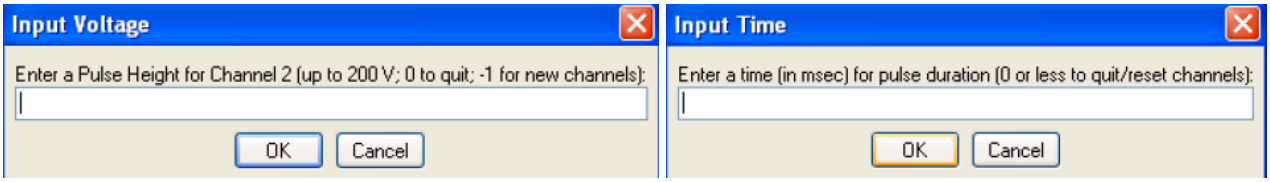


- Click "Reset Channels" to re-generate the original channel selection dialog, where all channels are available again.

Once the channels have been specified, the program proceeds to a dialog box that allows you to set the channel voltage(s). Specification prompts begin with the lowest channel selected—the order in which channels were selected is ignored (for instance, if Channel 3 was

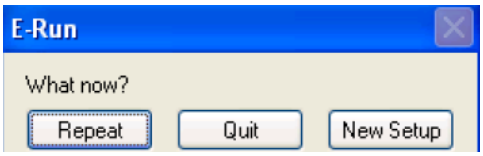
selected before Channel 1 was selected, the first "Input Voltage" dialog would still prompt the user to specify the voltage for Channel 1).

The figure below left illustrates the "Input Voltage" dialog. Clicking "Cancel" or leaving the edit box blank and clicking "OK" will provide the option of quitting or going back to respecify the channels. Entering -1 into the text box will directly allow channel respecification, while entering zero will cause the program to quit with no additional prompts.



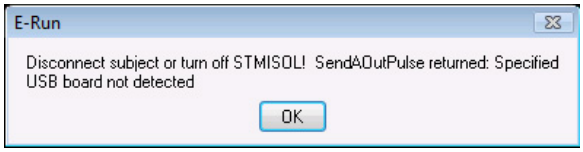
After the voltage(s) have been established, the program allows the user to specify the duration of the pulse. The figure above right illustrates this dialog. Entering a number will cause E-Prime® to send the command to the USB1208HS, so if output will be recorded on another device, that recording should be started just before "Enter" is typed or "OK" is clicked. Entering a value less than or equal to zero will generate a prompt to quit or return to channel selection.

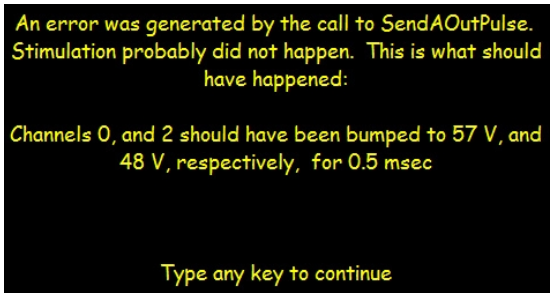
After the pulse(s), an information screen similar to the image depicted below left will be generated to indicate what the device just did. If recording the outputs, this is a useful time to verify amplitudes and timing. As instructed, you may type any key to continue. A prompt as indicated below right will then appear to provide the user with three options.



Click "Repeat" to produce another stimulus with the exact same parameters or "Quit" to terminate the program. Click "New Setup" to specify all parameters again, starting with the channel selection dialog.

If an error occurs, then instead of the above sequence, a dialog will alert the user of the problem with some diagnostic information such as shown below. The dialog box will be followed by a message confirming the settings sent by the user, such as at right.





When quitting the program used with STMISOC, another dialog box appears to direct the user to disconnect the subject. This allows the program to call SetStimulatorZero again to re-set the USB1208HS voltage to zero volts without presenting the subject with any unwanted transient voltage spikes.

Three other user interface examples, USB1208HS_AOutSimple and USB1208HS_AOutSine, and StimAndShow are included with the distribution CD. Their usage is documented separately in Readme files in the folders that contain the programs.

Recording

To verify operation of the USB1208HS, the analog output channels can be monitored. Many hardware setups can be used, such as an MP150 and a UIM100C.

1. The UIM100C connects with miniature mono phone plugs (3.5 mm TS connectors).
    a. The sleeve of the phone plug should electrically connect to the analog ground (AGND) terminal of the USB1208HS (there are two AGND terminals on the 4AO model; they are equivalent to each other).
    b. The tip of the phone plug should be electrically connected to the channel that will be measured (e.g., AOUT0).
2. Acq*Knowledge*® graph template files FourChannelOut.gtl and OneChannelOut.gtl are included with the sample programs. These files will configure Acq*Knowledge*® to read channels 1 through 4 or just channel 1 respectively

on an MP150. If recording from more than one but fewer than four channels, use FourChannelOut, select "MP150 > Set Up Channels...", and uncheck the channels that will not be used.

- Channel scalings in these gtl files map 10 V to 200 V (for STMISOL/STMISOLA) or 10 V to 100 V (for STMISOC in 1:10 mode), so recording from the USB1208HS analog outputs will report the voltages that an appropriately configured stimulator should produce when connected to these analog output channels; the values reported by Acq*Knowledge*® will correspond quantitatively to the voltages input to SendAOutPulse.

Optional Calibration

The USB1208HS uses 12-bit digital to analog converters. The assumption of "SendAOutPulse" is that digital values 0 through 4095 correspond to potentials of -10 V through 10 V respectively (note these numbers correspond to the USB1208HS analog outputs directly, not after having been scaled upward through a stimulator). "SendAOutPulse" as written should suffice for most applications.

- In BIOPAC validation tests, USB1208HS Channel 0 configured to hold a DC level of 10 V was measured as 10.00 V on a digital voltmeter.

To test device calibration, check the levels of two pulses of different input voltages, and change the scaling in SendAOutPulse. For example, if USB1208HS_AOut is run while measuring the outputs with an MP150 as described in the previous section and if under these conditions software inputs of 20 V and 75 V produce responses of 19 V and 74.5 V, respectively, the following steps should be performed:

1. Determine the digital values sent to the USB1208HS. The relevant line in the software (for STMISOL/STMISOLA) is:

   DACounts(k) = CInt(FullVoltages(k)*DAZero/200 + DAZero)

   DAZero for the USB1208HS is 2048, so the digital values in this example are:

   round(20*2048/200 + 2048) = 2253, and round(75*2048/200 + 2048) = 2816

2. Construct a new equation mapping desired output to calibrated input.

   2253 corresponds to 19 V and 2816 corresponds to 74.5 V, so a new equation mapping voltages to digital values would be:

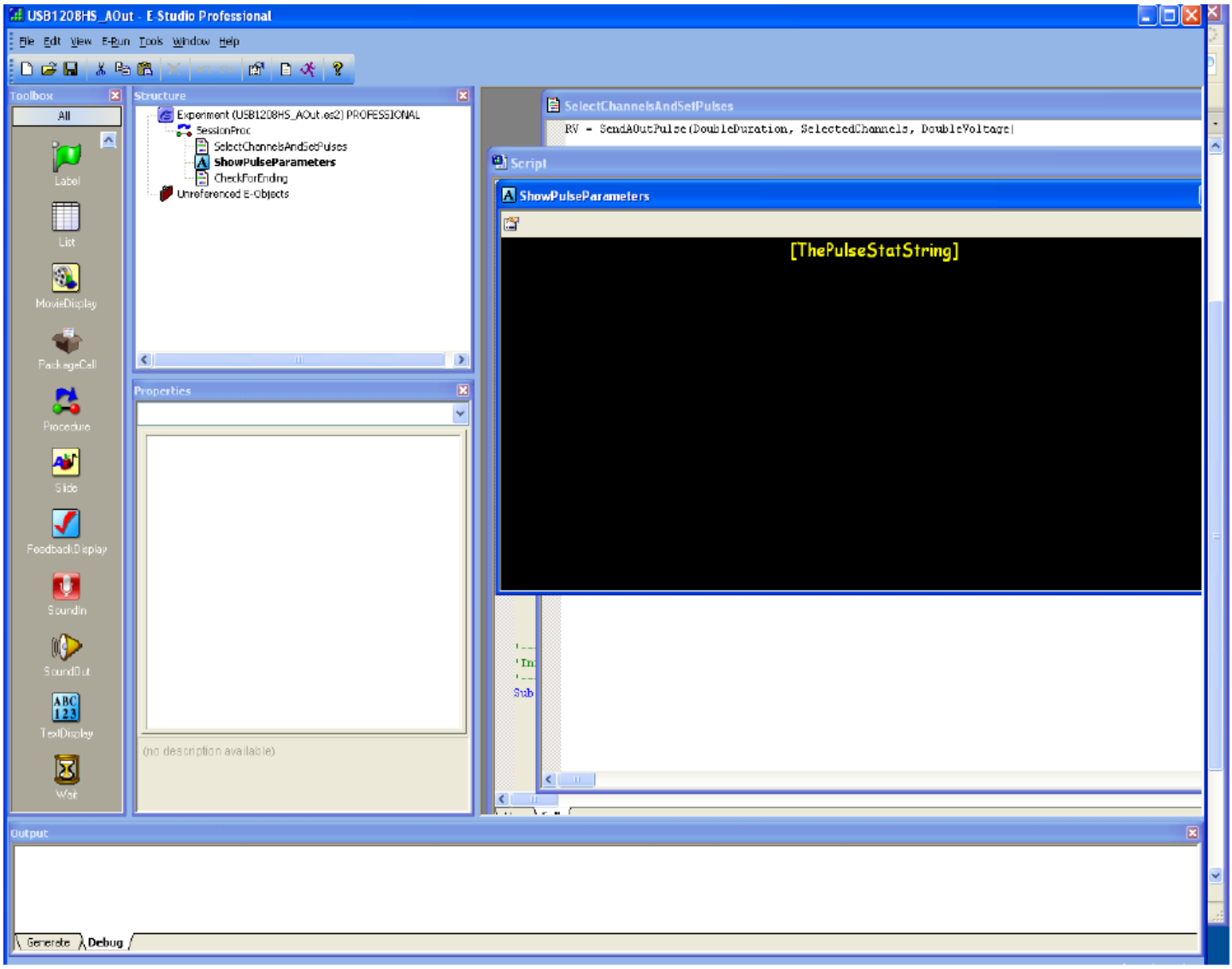
   DInput = 10.14*VOut + 2060

3. Replace the mapping in USB1208HS_AOut. The new line should read:

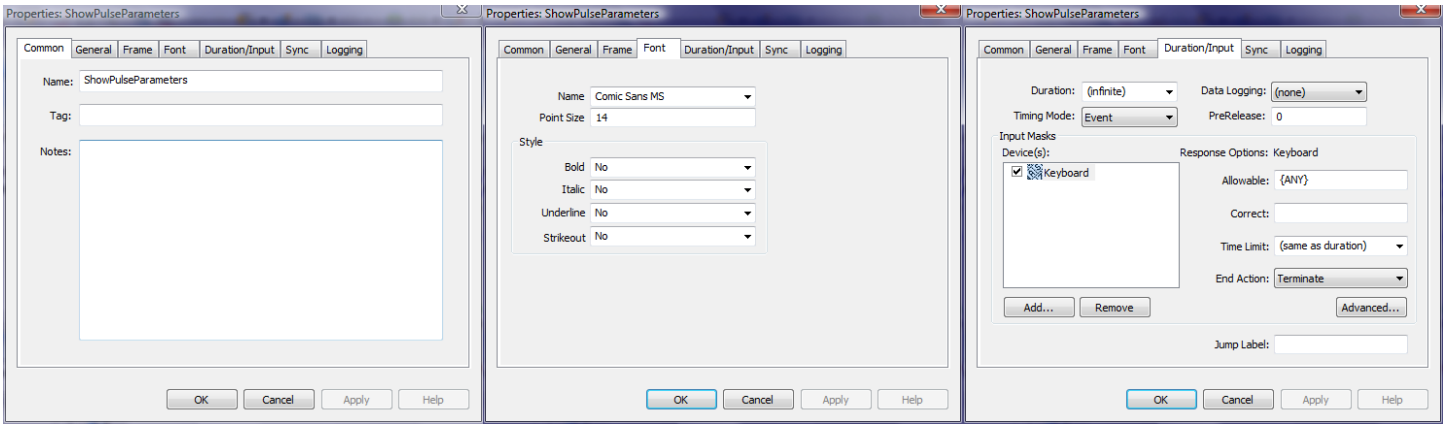
   DACounts(k) = Cint(FullVoltages(k)*10.14 + 2060)

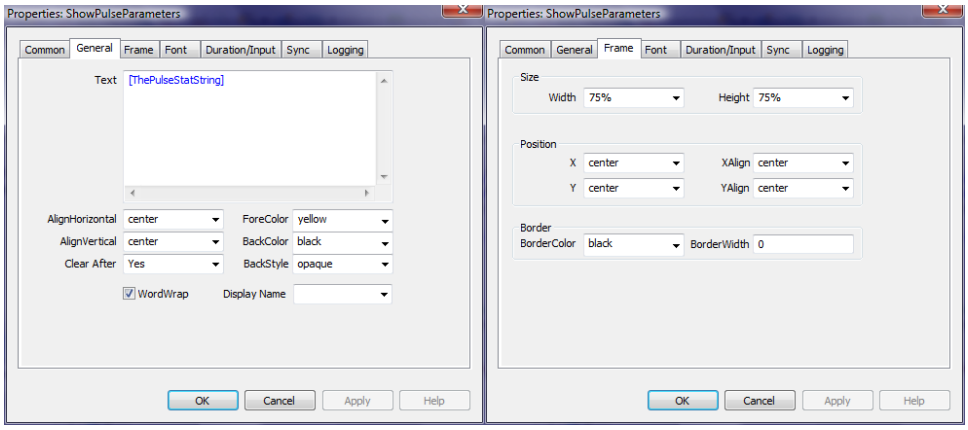
Running Test Software with Older Versions of E-Prime®

E-Prime® Version 1 is not able to read USB1208HS_AOut. With minor editing, users should be able to reconstruct the file in version 1 format, however. The structure of the program is indicated in the tree shown in the figure below.



Note that the program consists of the User Script and two InLine script modules sandwiching a Text Display object. The text display object contains only the line “[ThePulseStatString]” as indicated in the figure above. Properties of “ShowPulseParameters” are indicated in the figures below. The contents of the User Script are included in the file UserScript.txt. The other two inline scripts are contained in SelectChannelsAndSetPulses.txt and CheckForEnding.txt.




Properties: ShowPulseParameters
Common
General
Frame
Font
Duration/Input
Sync
Logging
Text
[ThePulseStatString]
AlignHorizontal
center
ForeColor
yellow
AlignVertical
center
BackColor
black
Clear After
Yes
BackStyle
opaque
WordWrap
Display Name
OK
Cancel
Apply
Help
Properties: ShowPulseParameters
Common
General
Frame
Font
Duration/Input
Sync
Logging
Size
Width
75%
Height
75%
Position
X
center
XAlign
center
Y
center
YAlign
center
Border
BorderColor
black
BorderWidth
0
OK
Cancel
Apply
Help