Sutter Instrument
Integrated Patch Amplifier Application Programming Interface
IPA API

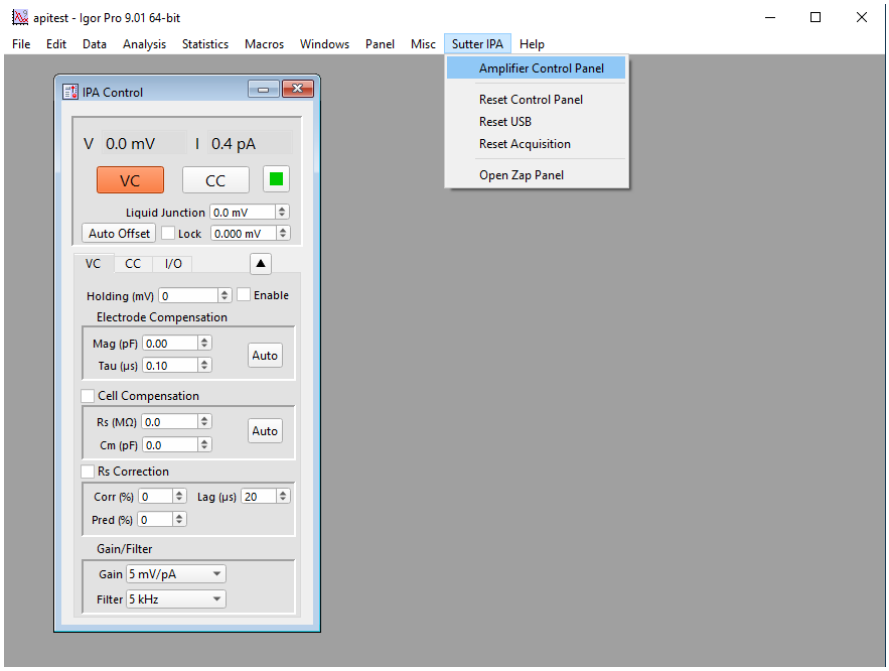
For use with Wavemetrics Igor Pro version 8 or 9

Package Contents:

- SutterXOP_Win-64.xop
- SP_IPA_ControlPanel (standalone).ipf
- SP_Drivers folder
- Example procedure code
- This document

Initial Setup:

1. Install driver for IPA amplifier (run SP_Driver_Installer.exe)
2. Load SutterXOP_Win-64.xop into your C:\Program Files\WaveMetrics\Igor Pro 9 Folder\Igor Extensions (64-bit) folder.
3. Load SP_IPA_ControlPanel (standalone).ipf into your C:\Program Files\WaveMetrics\Igor Pro 9 Folder\Igor Procedures folder.
4. When you start Igor, a menu will be added to Igor called Sutter IPA. From here, you can open the amplifier control panel.



Amplifier Control:

1. Opening the amplifier control panel will automatically connect the amplifier to Igor. If the amplifier is turned off, or loses the USB connection, the green light on the control panel will turn red. The USB can be reset either by pressing the red button or from the Sutter IPA menu.
2. Communication with the amplifier can be either directly through the control panel or programmatically, by using the *IPA_SetValue* command as described in Appendix A.
3. Amplifier settings can be read using the *IPA_GetValue* command. Keywords are defined in Appendix B. If a number of amplifier settings need to be read at once, we have included the shell function called *IPA_ReadStructure*.
4. When amplifier settings are changed, a shell function called *IPA_meta* is called. This allows logging of any changes in the amplifier during an experiment. Note: the labels for the metadata do not directly correspond with the keywords for the *IPA_GetValue* command. Please contact Sutter Instrument for a list of metadata labels.
5. We have added a Zap function, which can be accessed from the SutterIPA menu. In addition, we have a Buzz function, used when the amplifier is in current clamp. This can be accessed using the *IPA_Buzz(n)* function (where *n* is duration of buzz in ms). Finally, we also have a seal test function available through the *IPA_SetValue* command (Appendix A). However, this seal test is not functional on headstage 2 of the dIPA. If you require a seal test for the dIPA, please contact Sutter Instrument for a software solution.

Data Acquisition:

1. Data acquisition is controlled by SutterDAQ XOP functions as described in Appendix C. The following commands are most used: *SutterDAQScanWave* to collect data, *SutterDAQWriteWave* to provide output stimulation, *SutterDAQClock* to control the onboard timer (for clock-triggered acquisition), and *SutterDAQReset* for stopping and resetting data acquisition. A simple example procedure file is included in Appendix D.
2. If the user prefers to utilize a 3rd party data acquisition board, the IPA amplifiers have analog outputs. The SIGNAL OUTPUT is a switching 'primary' output (current in VC; voltage in CC). This output is scaled based on the gain setting of the IPA control panel. The COMMAND MONITOR is the secondary output (voltage in VC; current in CC). If it is voltage, the output is scaled at 10mV/V. For current, it is scaled at 2 pA/mV. The COMMAND IN provides the headstage stimulus, which adds to any signal provided by the amplifier control panel. Command stimuli in voltage clamp should be scaled at 10V/V and for current clamp should be scaled at 0.5 mV/pA.

Appendix A. The IPA_SetValue command

The IPA_SetValue command provides programmatic control over the IPA amplifier. It utilizes the format: IPA_SetValue*(HS, keyword, value) where:*

HS: 0 for currently selected headstage, or 1-based if multiple amplifiers (or double IPA). For a single headstage IPA, either 0 or 1 will produce the same effect.

keyword: As listed below.

value: Depends upon keyword—see below.

Keywords:

"Reset" : Resets all amplifier controls to default. (HS is ignored).

"SelectHS" or "SelectProbe": Sets amplifier panel to HS# (1 based). This has no effect if only a single IPA is connected.

"CCMode": Sets headstage to Current Clamp (*value* is ignored)

"VCMode": Sets headstage to Voltage Clamp (*value* is ignored)

"IHold": Sets the holding current in Current clamp to *value* (checkbox is automatically enabled when amplifier is in CC mode).

"IHoldOn": Enables/disables holding current in CC mode.

"VHold": Sets the holding potential in Voltage clamp to *value* (checkbox is automatically enabled is in VC mode).

"VHoldOn": Enables/disables holding potential in VC mode.

"Filter", "IFilter" or "VFilter": Sets the amplifier filter setting (same for CC and VC). Possible settings are (500;1kHz;2kHz;5kHz;10kHz;20kHz). Function will set to the closest setting given an actual value. Thus, either 5000 or 4900 will set the filter to 5kHz.

"VGain": Sets the input gain in Current clamp mode. Possible settings are (10;20;50;100;200;500 mV/mV).

"IGain": Sets the input gain in Current clamp mode. Possible settings are (0.5;1;2.5;5;10;25 mV/pA).

"Offset": Sets the offset (in Volts).

"OffsetLock": Toggles the offset lock on the control panel (1 = locked).

"AutoOffset": Calls the automatic offset control (offset lock must be disabled)

"ECompMag": Sets electrode compensation magnitude (SI units, so *value* is between 0 and 25e-12.

"ECompTau": Sets electrode compensation tau value (SI units, between 0.1e-6 and 4.5e-6).

"RsComp": Sets the whole-cell compensation Rs value (between 0 and 100e6). Cell compensation checkbox must be enabled.

"CmComp": Sets the whole-cell compensation Cm value (between 0 and 100e-12). Cell compensation checkbox must be enabled.

"RsCompOn": Enables/disables Cell compensation.

"RsPred": Sets the series resistance prediction value (% between 0 and 1). Rs correction checkbox must be enabled.

"RsCorr": Sets the series resistance correction value (% between 0 and 1). Rs correction checkbox must be enabled.

"RsCorrOn": Enables/disables Rs correction.

"RsLag": Sets the series resistance correction lag time (between 0 and 200e-6). Rs correction checkbox must be enabled.

"Bridge": Sets current clamp bridge balance (between 0 to 200e6). Bridge balance checkbox must be enabled.

"BridgeOn": Enables/disables bridge balance.

"AutoEComp": Activates the automatic electrode compensation routine.

"AutoCellComp": Activates the automatic cell compensation routine.

"DynHold": Sets the dynamic hold voltage (between -1 and 1 V) for slow voltage control in CC mode. Dynamic hold checkbox must be enabled.

"DynHoldOn": Enables/disables dynamic hold.

"ECompOn": Enables/disables electrode compensation (capacitance neutralization) in current clamp mode.

"DigOutWord": Sets the digital outputs.

"AuxOut1": Sets the auxiliary output #1 (between -10 and 10V).

"AuxOut2": Sets the auxiliary output #2 (between -10 and 10V).

"Reconnect": Resets the USB connection.

“SealTest”: Sets the HS#1 seal test. Value=0 off; otherwise value is amplitude in mV (scaled 20 pA/unit in CC). SealTest automatically turns off if the amplifier is switched from VC to CC (or vice versa), or if the headstage tab is changed (on the dIPA or multiple amplifiers). Seal test has a 100ms duration which cannot be changed. *Note: the seal test is not functional on headstage 2 of the dIPA. If you require a seal test for the dIPA, please contact Sutter Instrument for a software solution.*

"AutoBridge": Not currently implemented.

Appendix B. The IPA_GetValue command

The IPA_GetValue command provides the ability to read the amplifier settings. It utilizes the format: IPA_GetValue*(HS, keyword) where:*

HS: 0 for currently selected headstage, or 1-based if multiple amplifiers (or double IPA). For a single headstage IPA, either 0 or 1 will produce the same effect.

keyword: as listed below.

All values are returned in SI units (e.g., 1e-12 A not pA).

Keywords:

"ActiveIPA": In the case of multiple amplifiers, returns the currently loaded amplifier (1-based), otherwise returns 1.

"ActiveHS" or "ActiveProbe": Returns active headstage number (1-based).

"AmplIndex": Returns the index (0-based) for the active amplifier.

"HSIndex": Returns the index (0-based) for the active headstage.

"NumAmps": Returns the total number of amplifiers connected.

"NumHS" or "NumProbes": Returns the total number of headstages connected.

"AmpType": Returns the amplifier type (1=IPA, 2=dIPA, 3=Dendrite)

"AmpSN": Returns the amplifiers serial number.

"CCMode": Returns 1 if the amplifier is in current clamp mode.

"VCMode": Returns 1 if the amplifier is in voltage clamp mode.

"Filter" or "IFilter" or "VFilter": Returns the current low-pass filter setting.

"VGain": Returns the gain of the voltage signal. If in voltage clamp, returns 10V/V.

"IGain": Returns the gain of the current signal. If in current clamp, returns 5e8 V/A (0.5 mV/pA).

"OffsetLock": Returns 1 if offset lock is checked.

"Offset": Returns current offset.

"LJP": Returns current liquid junction potential setting.

"DynHoldOn": Returns 1 if dynamic holding is checked.

"DynHold": Returns current dynamic holding potential.

"ECompMag": Returns the electrode compensation magnitude.

"ECompTau": Returns the electrode compensation tau.

"ECompOn": Returns 1 if capacitance neutralization (in CC) is activated.

"RsCompOn": Returns 1 if whole cell compensation is activated.

"RsComp": Returns whole cell compensation resistance value.

"CmComp": Returns whole cell compensation capacitance value.

"RsCorrOn": Returns 1 if series resistance correction is activated.

"RsPred": Returns series resistance prediction %.

"RsCorr": Returns series resistance correction %.

"RsLag": Returns the series resistance correction lag value.

"Bridge": Returns the bridge balance value.

"BridgeOn": Returns 1 if the bridge balance is active.

"IHoldOn": Returns 1 if the holding current is activated.

"IHold": Returns the holding current (0 if holding current is not activated).

"VHoldOn": Returns 1 if the holding potential is activated.

"IHold": Returns the holding potential (0 if holding potential is not activated).

"IMon": Returns the amplifier current reading.

"VMon": Returns the amplifier voltage reading.

"DigOutWord": Returns the state of the 8 digital outputs.

"AuxOut1": Returns the setting of auxiliary output #1.

"AuxOut2": Returns the setting of auxiliary output #2.

"AuxOut3": Returns the setting of auxiliary output #3 (Dendrite only).

"AuxOut4": Returns the setting of auxiliary output #4 (Dendrite only).

"AuxIn1": Returns the reading from auxiliary input #1.

"AuxIn2": Returns the reading from auxiliary input #2.

"AuxIn3": Returns the reading from auxiliary input #3.

"AuxIn4": Returns the reading from auxiliary input #4.

"AuxIn5": Returns the reading from auxiliary input #5 (Dendrite only).

"AuxIn6": Returns the reading from auxiliary input #6 (Dendrite only).

"AuxIn7": Returns the reading from auxiliary input #7 (Dendrite only).

"AuxIn8": Returns the reading from auxiliary input #8 (Dendrite only).

Appendix C. SutterXOP-64 Functions and Operations

SutterXOP provides the interface between Igor Pro and Sutter Instruments Integrated Patch Amplifier (IPA). It adds the following functions and operations to Igor (updated 12/27/22).

Most of these functions are handled by the IPA control panel. For example, the user should not need to directly manage the USB connection, make reads to the amplifier (SutterDAQread), or write directly to the amplifier (SutterDAQwrite).

SutterDAQUSBReset()

Resets the USB connection for attached amplifiers.

SutterDAQUSBReset closes all USB connections and then reopens them. Function returns the number of devices that have been connected. Errors return negative number.

SutterDAQConnect(*n*)

Opens USB connections

SutterDAQConnect opens the USB connection between Igor Pro and any connected IPAs. Function returns the number of devices that have been connected. Errors return negative number. *Note: Deprecated—users should use SutterDAQUSBReset to open connections to amplifier, which manages previously open connections prior to connecting.*

SutterDAQUSBClose()

Closes USB connections.

SutterDAQUSBClose closes all USB connections between Igor Pro and any attached IPAs.

SutterDAQSN(*n*)

Returns amplifier serial number.

SutterDAQSN returns a string containing the serial number of the selected amplifier (n = 0 if only one amplifier is connected).

SutterDAQRead(*n1, n2*)

Returns selected information from an attached amplifier. *Note: it is advised to use the IPA_GetValue function to read amplifier information.*

SutterDAQRead returns information about the selected amplifier's current state. *n1* is the amplifier number (0 if only one amplifier is connected). n2 is the item number:

0: Gain (byte 0) and VC vs CC (byte 1, i.e., VC adds 256), HS#1
1: Primary input reading (Current in VC; Voltage in CC), HS#1
2: Secondary input reading (Voltage in VC; Current in CC), HS#1
3: Auxiliary input #1 reading
4: Auxiliary input #2 reading

5: Auxiliary input #3 reading (for single IPA, #1 for dIPA)
6: Auxiliary input #4 reading (for single IPA, #2 for dIPA)
7: Low Pass Filter Setting (20kHz=0; 10kHz=1; 5kHz=2; 2kHz=3; 1kHz=4; 500Hz=5), HS#1
8: Output waveform buffer availability (used during data acquisition)
9: Electrode compensation (magnitude) in bits (*25pF/16384), HS#1
10: Electrode compensation (phase) in bits (*100%/16384), HS#1
11: Cell Compensation tau (Rs*Cm) in bits (*1.328s), HS#1
12: Cell Compensation Cm value in bits (*100pF/16384), HS#1
13: Time elapsed since last trigger (in seconds)
14: Time elapsed since amplifier was turned on (in seconds)
15: dIPA HS#2: Gain and VC vs CC
16: dIPA HS#2: Electrode compensation (magnitude) in bits (*25pF/16384)
17: dIPA HS#2: Electrode compensation (phase) in bits (*100%/16384)
18: dIPA HS#2: Cell Compensation tau (Rs*Cm) in bits (*1.328 s)
19: dIPA HS#2: Cell Compensation Cm value in bits (*100pF/16384)
20: *reserved*
21: dIPA HS#2: Primary input reading (Current in VC; Voltage in CC)
22: dIPA HS#2: Secondary input reading (Voltage in VC; Current in CC)
23: Auxiliary input #3 reading for double IPA
24: Auxiliary input #4 reading for double IPA
38: Firmware version number.

SutterDAQReadClock(*n1,n2*)

Reads onboard timer from selected amplifier

SutterDAQReadClock returns a precise time from the selected amplifier*. n1* is the amplifier number (0 if only one amplifier is connected). *n2* determines timer reading:

0: Current time
1: Time of last trigger
2: Elapsed time since last trigger
3: Resynch. This resynchronizes the amplifier on time to a current computer timestamp

SutterDAQClock(*n1,n2,n3*)

Sends a timer command to the amplifier

SutterDAQClock sends a clock command to the amplifier. *n1* determines if this is a single (0) trigger mode or continuous mode (continues until stopped by user). *n2* is the timer period (accurate to 0.1 s). *n3* determines if trigger should happen at the end of the timer period (0) or at the beginning (1).

Example: SutterDAQClock(1,10,1) will create a trigger every 10 seconds until halted by the user.

if multiple amplifiers are connected, the clock command is sent to amplifier 0. The second amplifier should be triggered by connecting the TRIGGER OUT from the first amplifier to TRIGGER IN on the second.

SutterDAQTrigger(*n1*)

Sends an arm external trigger command to the amplifier

SutterDAQtrigger sends an arm trigger command to the amplifier. if n1 = 1 the external trigger is armed, if n1= 0 the external trigger is disarmed.

SutterDAQWrite(*n1,n2,n3,n4,n5*)

Writes a packet to the selected amplifier

SutterDAQWrite sends a four character packet to the selected (*n1*) amplifier. Please contact Sutter Instrument technical support for further information. Amplifier commands should be made using the IPA_SetValue function.

SutterDAQReset()

Resets the amplifier A/D functions.

SutterDAQReset resets the A/D functions on the amplifier (amplifier #0 if multiple amplifier are connected). This stops data acquisition and waveform generation, resets the onboard timer, as well as closing any data acquisition threads in the XOP.

SutterDAQScanWave /T[=*t*] /B[=b] /C[=c] /MULT[=m] /G=gainwave /H=*hookstring* /E=*errorstring* Inputwavelist

The SutterDAQScanWave operation acquires data from the amplifier

SutterDAQScanWave an operation for streaming data into the computer from the amplifier. It collects data into waves as listed in the Inputwavelist text wave (described below). The sampling rate and number of samples are derived from wave referenced in the first row of InputWavelist. With a maximum sampling rate of 50 kHz. **Note: All input waves must have the same x scaling and same number of points.**

All inputs are automatically scaled as they are collected. Additional scaling and offsets can be applied during acquisition using the optional /G flag as described below.

For triggered acquisition, the trigger time is written to root:SutterIPA:SD_sweeptime. Time is maintained on amplifier and is synchronized to the computer when the USB connection to the amplifier reset.

Flags:

/T[=*t*] Trigger mode for input:

t=0 trigger occurs immediately (if /T is omitted, same as /T=0).
t=1 triggered by clock trigger (same as /T).
t=2 external (BNC) trigger.

/B[=*b*] Background mode:

b=0 control is not returned to Igor until the entire sweep is collected (if /B is omitted, same as /B=0).
b=1 a separate thread is created to acquire data. Control is returned to Igor immediately (recommended). (same as /B).

Note: Current XOP ONLY works in background mode. /B is presumed. Contact Sutter Instrument technical support for further information or if you require data acquisition to occur directly in the main thread.

/C[=*c*] Continuous mode:

c=0 Normal acquisition (if /C is omitted, same as /C=0).
c=1 Continuous acquisition. The next sweep begins with the next sample after the previous wave is filled (same as /C).
Note: Depending on the sweep length and sampling rate, this mode can tax Igor's resources. While the endhook function is called between sweeps, users should limit what is performed by Igor during this time.

/MULT [=m] Multiple amplifier support

m=0 Single amplifier mode (if /MULT is omitted, same as /MULT = 0)
m=1 Multiple amplifier mode (same as /MULT)
If only one amplifier is connected, this flag is ignored.

Contact Sutter Instrument technical support for further information on controlling multiple amplifiers.

/G=gainwave

Optional pass a wave reference that contains a two dimensional wave that corresponds to the Inputwavelist (described below). For each channel (row) collected, the first column provides an additional gain factor and the second column provides an offset. *Note: a gain factor of 0 is ignored.*

/H=*hookstring*

String that is executed when thread has finished sending collecting the sweep. This allows you to call a function to process the acquired data and retrigger acquisition.

/E=*errorstring*

String that is executed if an error occurs in the background thread.

The Inputwavelist must be a two-dimensional text wave. The first column provides the name (with full path) of the wave which will collect the data. The second column is the channel number of the input:

0 is the input current (in both VC and CC) for HS#1
1 is the input voltage (in both VC and CC) for HS#1
2 is the input current (in both VC and CC) for HS#2 on a double IPA
3 is the input voltage (in both VC and CC) for HS#2 on a double IPA
2-5 are the auxiliary analog input channels for single IPAs
4-7 are the auxiliary analog input channels for double IPAs

Additional columns are currently ignored by the XOP unless you are using multiple amplifiers. Contact Sutter Instrument for more information.

SutterDAQWriteWave /T[=*t*] /R[=*r*] /RHP[=*rhp*] /LV[=*lv*] /A[=*a*] /CC[=*cc*] /MULT[=*m*] /H=*hookstring* Outputwavelist

The SutterDAQWriteWave operation sends data to the amplifier's D/A converter

SutterDAQWriteWave is an operation for streaming data out to the amplifier which are written to the headstage, the digital outputs and the analog outputs. It shares the triggering mechanism with SutterDAQScanWave.

Flags:

/T[=*t*] Trigger mode for output:

t=0 Data starts streaming immediately (if /T is omitted, same as /T=0).
t=1 Data outputs are triggered by clock trigger (same as /T).
t=2 Data outputs are triggered external (BNC) trigger.

/R[=*r*] Repeat mode:

r=0 no repeat (if /R is omitted, same as /R=0).
r=1 triggered repeat. Waves are repeated on next trigger (same as /R).
r=2 continuous repeat. Wave are repeated as soon as output is complete.

Note: For repeat mode, output waves must be less than 4096 samples/channel.

/RHP[=*rhp*] Relative to holding potential:

rhp=0 amplifier writes absolute values (if /RHP is omitted, same as /RHP=0).
rhp=1 amplifier writes values relative to holding potential for the channel (same as /RHP).

Note: RHP can be set independently for each headstage. Simply bitshift the value for HS#2.

/LV[=*lv*] Last value:

lv=0 amplifier returns to holding potential after output is complete (if /LV is omitted, same as /LV=0).
lv=1 amplifier remains on last value of output wave (same as /LV).

/A[=*a*] Append to previous (requires that amplifier has not completed previous output).

a=0 no append (same as omitting /A).
a=1 append to previous output. (same as /A).

Note: This functionality has not been extensively tested.

/CC[=*cc*] Current clamp mode

Command potential wave is converted into a current stimulus (assumes that output wave is scaled in Amps). Note: CC flag can be set independently for each headstage. Simply bitshift the value for HS#2.

/MULT [=m] Multiple amplifier support

m=0 Single amplifier mode (if /MULT is omitted, same as /MULT = 0)
m=1 Multiple amplifier mode (same as /MULT)

If only one amplifier is connected, this flag is ignored. **Contact Sutter Instrument technical support for further information on controlling multiple amplifiers.**

/H=*hookstring* String that is executed when thread has finished sending data to amplifier

(NOTE: this is not the same time as when the amplifier has finished outputing the data. In SutterPatch, this hook is unused. Instead, software is controlled from the scan wave hook.

The Outputwavelist must be a 2-dimensional text wave. The first column provides the name (with full path) of the output wave. The second column is the channel number of the output:

	single IPA	double IPA
Command potential HS#1	0	0
Command potential HS#2	x	1
Auxiliary output #1	1	2
Auxiliary output #2	2	3
Digital output (8 bit word)	3	4

The sampling rate and number of samples are derived from wave referenced in the first row of OutputWavelist, with a maximum output sampling rate of 10kHz Note: All output waves must have the same scaling and same number of points.

Appendix D. Example Data Acquisition Procedure

The following procedure provides an example of the SutterDAQ data acquisition flow. This basic example collects both the current and voltage inputs from an IPA amplifier. 10 sweeps are collected and stored in 2-d data waves. This example can be copied and pasted into an Igor procedure window. Activate by calling SetupandStart().

```
Function SetupandStart()
// I. Make input waves
    //create both current and voltage waves
    Make/d/o/n=(1000) root:input_i, root:input_v  //200ms @ 5kHz
    Wave input_i = root:input_i
    Wave input_v = root:input_v
    input_i = nan
    input_v = nan
    Setscale /p x, 0, 0.0002, "s", input_i, input_v
    Setscale d -20e-9,20e-9, "A", input_i
    Setscale d -1,1, "V", input_v

// II. Make inputwavelist
    Make /O/T /n=(2,2) root:InputWaveList
    Wave /T InputWaveList = root:InputWaveList
    InputWaveList[0][0] = "root:input_i"
    InputWaveList[1][0] = "root:input_v"
    InputWaveList[0][1] = "0"
    InputWaveList[1][1] = "1"

// III. Set key variables
    Variable /G root:SweepNumber=0
    Variable ISI = 2

// IV. Display Waves - a simple graph to display input waves
    Display /W=(50,50,450,400) input_i
    AppendToGraph/L=leftv input_v
    ModifyGraph lblPos(left)=50,lblPos(leftv)=50
    ModifyGraph freePos(leftv)=0
    ModifyGraph axisEnab(left)={0.55,1}
    ModifyGraph axisEnab(leftv)={0,0.45}

// V. Make output stimulus
    Make /d/o/n=200 root:stim_v
    Wave stim_v = root:stim_v
    Setscale /p x, 0, 0.001, "s", stim_v
    //Give it a -10 mV step
    stim_v = 0
    stim_v[50,150] = -0.010

// VI. Make outputwavelist
    Make /O/T/N=(1,2) root:OutputWaveList
    Wave /T OutputWaveList = root:OutputWaveList
    OutputWaveList[0][0] = "root:stim_v"
    OutputWaveList[0][1] = "0"

// VII. Start acquisition
    // First send stimulus, which will be repeated each time, so only need to send once.
    SutterDAQWriteWave /T=1  /R=1 /RHP=1  OutputWaveList
    // Next request triggered acquisition
    SutterDAQScanWave /T=1  /B=1 /H="ContinueExample()" root:InputWaveList
    // Finally, trigger acquisition using ISI listed above.
    SutterDAQClock(1,ISI,1)
End
```

```
Function ContinueExample()
// This function is automatically called when the input waves have been filled.
// We will increment our sweep counter, copy to a data wave, and rearm acquisition.

    NVar SweepNumber = root:SweepNumber
    if (SweepNumber == 0)
        //for simple example we'll overwrite data
        Make /o/n=(1000,10) mydata_i, mydata_v
        Setscale /p x, 0, 0.0002, "s", mydata_i, mydata_v
    endif
    Wave input_i = root:input_i
    Wave input_v = root:input_v
    mydata_i[][SweepNumber] = input_i[p]
    mydata_v[][SweepNumber] = input_v[p]

    //Increment SweepNumber
    SweepNumber += 1

    //We could do real-time analysis on the data here...

    //Check if we should end
    If (SweepNumber == 10)
        //Reset the amplifiers acquisition (stops onboard clock)
        SutterDAQReset()
        Return 0
    else
        //Otherwise rearm acquisition
        SutterDAQScanWave /T=1  /B=1 /H="ContinueExample()" root:InputWaveList
    endif
End
```