PsychLab PCC
PCC Stimulus Control Language reference.
WWW.PsychLab.COM

June 2009.

Section 1
General description

PsychLabAcquire software is specifically designed to work with the problems encountered in measuring data in the context of Psychophysiology experimentation. It is pitched to suit academic users who will appreciate features which, while requiring more effort to configure, provide the flexibility to accommodate unusual needs without the limitations found in other software systems. PCC (Psychlab Command Code) is built into SAM1 measurement software. This script defines each experiment and may be adapted by the user to control stimulus presentation and interact with hardware.

PsychLab8 off-line analysis system allows reduction and modification of data recorded from PsychLabAcquire, while retaining compatibility with data generated from older hardware. The program shows each stage graphically and provides conversion of volume physiology data into numeric tables that allow comparison of performance at different stages in the record. Its many functions include evoked response analysis; trend analysis; filtering; and data export for further analysis. PsychLab8 is capable of working with multiple channels of different data types simultaneously. The wave analysis algorithm allows automatic quantification of fluctuations; macro files (.MAC) sequence program operations; and the Results Format Program (.RFP) is an intelligent script file providing wide ranging control over the format of results data.

An essential aspect of the analysis system is that it generally requires an event channel generated at the time measurement. This is used to determine critical time periods to focus upon data in the continuous recording for analysis. For this reason it is essential that the measurement procedure correctly marks events, such that each trial category is uniquely identifiable in the event channel. The event channel uses 8 bits, which gives 255 different possible event categories. Constructive design of the measurement script file allows the analysis system to separately group relevant data aspects relative to trial type.

Software upgrades are currently provided free of charge. Users who bought older versions of our software still benefit from enhancements of current and future systems, which may be freely downloaded from the PsychLab.com internet site.

Section 2
PsychLabAcquire: Command code program structure.

Unlike BASIC, all program lines which are not a jump to LABEL must be positioned at least one space away from the left margin. Labels may be used as jump to points and at the head of subroutines. The label MUST be positioned against the far left margin, and is normally typed in upper case to enhance program readability.

No string handling is supported, except in the print ".." statement.

Upper and lower case may be used as required - the interpreter is not case sensitive. However, labels are typed in upper case by convention.

As in BASIC, statements may be placed on consecutive lines or several statements may be placed on one line separated by colons.

As in BASIC, any combination of letters and numbers may be used for variables, but a number may not be used as the first character for a variable descriptor.

In general, a space must be placed between each item on a line.

Complex math expressions may be used, however PsychLab only supports +, -, *, and / operators. Unlike BASIC, implicit brackets are not automatically positioned around multiplication or division in a complex equation, i.e. use X = A + (B * C).

All variables are 'floating point'. The %, !, and & operators have no effect. There is a limit of 78 characters line width.

Special Characters and Other Syntax Rules.

Single inverted comma '
The single inverted comma allows a remark on a line of PCC code. Anything after it is completely ignored by PsychLab. Remarks may be added to the end of a line by placing the ' symbol before the remark.

D**ouble inverted comma " "**
Double inverted commas are used to put a label in a $measure metacommand line. Always close quotes. Double inverted commas are also used to contain wording that is to be printed on screen. There must always be a closing pair of double inverted commas. Unless inverted commas are used, the word following print will be considered as a variable, and its value will be shown. Please refer to the PRINT command.

Comma , Space

Commas or spaces may be used to separate parameters. PsychLab counts the number of commas to find out which parameter is being referred to. Either a comma or a space or both may be used. Multiple spaces are ignored. For example:

```
setTone 3, 200, 64, 100, 2, 300    ' ears, level, event, freq, attack, durat.
```

This command recognises each parameter by its position after the command word SetTone

Case sensitive?

No. When PsychLab evaluates PCC lines, it first converts all letters to upper case. However, the editor allows both upper and lower case to be used to write PCC lines. This is used to help distinguish LABELS at the head of subroutines, which are put in upper case in all the examples given here.

Colon :
The colon may be used to separate statements on the same line e.g.

x = x + 1: y = y + 1

Logical Operators

> (greater than) < (less than) = (equal to)
Used in if...then... statements used to direct program flow;. <> (not equal to) does not work, two separate statements must be used to obtain this condition using > in one statement and < in the next.

Line Length

PsychLab measurement system allows maximum 24 discrete items, or 78 characters including spaces per line e.g.

```
if time = t1 then gosub STIM
```

This line is 7 items and about 29 characters (including spaces).

PCC program variables.

PsychLab variables are words used to represent numbers. Any variable contains the value 0 when PsychLab measurement begins unless a program statement deliberately changes it. Variables used in PCC code may use any number of characters to 'describe' them, all of which must be exactly the same each time the variable is used in the program. (Variables used in the PsychLab processing RFP programs are less positively identified, using only the first and last characters). One exception to this rule, however, is that the system is not 'case sensitive', it does not matter whether the letters in the character description are upper or lower case.

The 'description' characters used to represent a variable may be chosen to help remember what the purpose of the variable is in the particular program. For example, in a program which delivers a number of stimuli, there might be a variable named StimCount, which is increased by one every time a new stimulus is delivered. This variable could just as easily be called variable1 or V1, but then it would be necessary to remember what varaible1 does, as opposed to varaible2 etc. The use of upper case characters to break up the descriptor into its meaningful parts is recommended (e.g. the upper case S and C in StimCount). Spaces or other punctuation characters are not allowed, so this is a useful way to enhance readability of the program.

Any mix of letters and numbers may be used to describe a variable, as long as the first character is not a number. The variable descriptor may not be the same as any of the reserved words used for pre-defined functions.

For those with knowledge of computer languages, PsychLab PCC variables are only numeric, floating point (meaning that they may be fractional and have a range in the region of 10 to the power of +/-13).

Examples of use of variables and math.

Variables might be used as in the following examples. These examples are intended to suggest uses, and demonstrate syntax, but should not imply limitation to these uses. Notice that all lines in the PCC program except labels must start at least one space from the left.

```
variable1 = .5
```

Value zero point five (0.5) is assigned to 'variable1'. Regardless of whatever value it had before, after execution of this line, it will contain 0.5. Any other value, positive or negative could have been assigned. It is important to note that with fractional numbers, the leading zero should not be given.

```
trial = trial + 1
```

Every time program execution includes this line, 'trial' will increase by one; ‘trial’ is just a user variable, it is not a program function or reserved word.

```
if  index = 10 then gosub SUB1
```

The value of ‘index’ is used to determine when a conditional jump will occur. Assuming that ‘index’ is changed as measurement continues, perhaps by counting the number of times a stimulus is presented, then only when it becomes ten (perhaps on the tenth stimulus) will the jump to subroutine SUB1 occur (which might perhaps be a UCS which is presented only once). Notice that the label SUB1 is given in upper case. This is just a convention, which helps to separate the subroutine labels when reading the program. PsychLab is not case sensitive, it sees upper and lower case letters the same.

```
if isiPointer = lastTrial then store
```

This is similar to the last example, except that isiPointer is compared to another variable ‘lastTrial’, and the operation that takes place when the condition is true is the ‘store’ function. ‘store’ is a reserved word, and its function is to end the recording.

The following are a few examples of more complex math expressions. Parentheses must be used where shown.

```
variable1 = variable1 + variable2
variable1 = variable1 + (variable2 * variable3)
```

In the next example, an expression is calculated separately before the result is used in the specific functions. These functions will not allow use of complex expressions directly for their parameters:

```
variable1 = variable2 + variable3
tone variable1
variable4 = table(variable1)
print variable4
```

The last examples demonstrate that the parameters used for 'tone' and 'print' must not be expressions, nor the subscript for 'table'. In order for the contents of the table to be used in either the tone or the print command, it must first be transferred to a variable. This is also true for other similar functions.

Reserved words

PsychLab measurement PCC language uses a number of reserved words which are recognised by the system and perform specific functions. For example, the word Tone is used to operate the TG2 tone generator. These words must therefore be avoided when choosing descriptions for user variables. The following is a summary of PsychLab PCC language reserved words. In addition, any word beginning with the $ sign.

BEEP	Make a sound in the computer
BIN8	Operate BIN8 TTL output
CLEAR	*Remove any picture currently showing in the media window
DELAY UNTIL	Wait at this statement until condition is true
EVENT	Read from (x = event) or write to (event = x) PsychLab event port
GOSUB ... RETURN	Jump program lines and return back
GOTO	Jump program lines
IF ... THEN ...	If expression after if is true, do ... after then
INT(..)	Convert fraction to whole number
LATENCY	Increases with time resolved to sampling interval
LOAD [Picture number]	*Load in a picture
NOISE	Operate white noise generator as configured by SETNOISE
PRINT	Write in the debug window
RANDOM	Produce random number between 0 and 255
RELAY	Operate relays on BIN8 device
RETURN	..from GOSUB
SETNOISE	Set up sound noise parameters in sound generator
SETTONE	Set up sound tone parameters in sound generator
SHOCK [level]	Operate shock generator
SHOW	*Show the loaded picture in the media window
STORE	End data acquisition
TABLE(..)	Array function (set up pseudo random conditions)
TIME	Returns absolute time since recording began
TONE	Operate tone generator

Section 3

PCC measurement program: Metacommands.

Metacommands have global effect on the configuration of the measuring system. They cannot be used in the dynamic section of the PCC program. They are normally grouped at the top of the PCC program, and used to set selected measurement channels, sample rate, etc. Metacommands are enacted when the PCC program is loaded, and when Main menu, PCC, Run metacommands now. is used.

Metacommand Summary

$RANDOM	Uniquely seed the random generation source.
$RATE [samples/sec]	Establish data acquisition sample rate
$PSYCHLAB	Used to set PsychLab amplifier controls, or disable a PsychLab unit.
$TABLE	Enter values into table array (like basic 'data')

$RANDOM

Causes values produced by the RANDOM command to be completely random. Without use of $RANDOM, the random function may produce the same sequence each time the SAM1 program is started.

$RATE [samples per second]

Sets the overall data rate used during data acquisition. Minimum PsychLab sample rate is 78, maximum depends on which amplifier/s are attached to the PsychLab unit. For example a single BioAmplifier will run at over 3kHz, but an EEG8 has a maximum of around 1.4kHz. Two EEG8s on one PsychLab unit has a maximum of 1024Hz. Setting sample rate with $RATE physically moves the slider of the acquisition sample rate control in the Properties window. When data are replayed, whatever rate is set by $rate is used by the analysis system to calculate time by counting the number of points and multiplying it by the interval between samples. It is possible to check whether the PsychLab unit is capable of maintaining the selected rate by looking at the number in the top left corner of each PsychLab window. This number is produced using the computer clock to count the number of samples and calculate sample rate. The number is inaccurate when sampling has only just started, because of buffering between PsychLab and the host computer, but it becomes more and more accurate as time progresses.

$PSYCHLAB [number], [hide], [range (range setting)], [highpass (high pass setting)], [low pass (low pass setting)], [hum (on or off)]

This metacommand may be used to assert required settings for PsychLab amplifiers, and to disable PsychLab units if their measures are not required. Multiple $PsychLab statements may be used for one PsychLab unit, or all the controls for that unit may be contained in a single $PsychLab statement. It is advisable to avoid very long lines in PCC programs, so it is recommended to set right and left controls using separate $PsychLab statements. This command and all PCC commands are not case sensitive. Commas or spaces may equally be used to separate items in the $PsychLab command line. There must be a space at the beginning of the line.

Examples:

```
$PSYCHLAB 1, range 100mV, highpass 30Hz, Lowpass 500Hz, hum on
$PSYCHLAB 1, range 200mV, highpass .1Hz, Lowpass 40Hz, hum off
$PSYCHLAB 2, hide
$PSYCHLAB 3, hum off
$PSYCHLAB 4, highpass 10Hz
```

[number]	-determines which PsychLab unit will be affected. It refers to the large number at back.
[hide]	-causes the particular PsychLab unit (in a multiple PsychLab system) to minimize, and no data to be stored for that unit.
[range (range setting)]	-sets the amplifier range for the selected side. mV or uV should be stated, there should not

be a space between the number and mV/uV.

[highpass (high pass setting)]	-sets the amplifier high pass for the selected side.
[lowpass (low pass setting)]	-sets the amplifier low pass for the selected side.
[hum (on or off)]	-sets the amplifier hum filter on or off for the selected side.

$TABLE

Used to define a series of values that can be accessed by the dynamic function **table(N)** . Each value following **$table** will be accessed as the **N**th value, starting at N = 1 . **$table** may be used on several different lines to avoid long lines of PCC. Values defined in the second **$table** line run consecutively from those defined in the first - e.g.

```
$table 7,19,4
$table 22,3
```

results in table element 1 containing value 7, element 2 value 19, element 3 value 4, element 4 value 22, etc.

Unused table elements contain 0. The table size may be up to 255 elements, each element value must be integer, between 0 and 32767. The first value defined by **$table** is accessed by **table(1)** in the dynamic section of the program.

Section 4

PCC measurement program: Dynamic commands.

Dynamic commands are used in the measurement control program to organise stimulus procedures. The language is like a reduced BASIC. Normally a 'closed program loop' is employed, such that the PCC program loops continuously. Branches may be made using the if … then statement. The language supports user variables, constants and has a number of functions relating to hardware. The dynamically interpreted section of the program is normally positioned below the metacommand section. A structure that gives rise to minimum complication employs a tight loop at the top of the program and subroutines listed underneath, organized such that most of the time (between stimulus trials) is spent in the MAIN loop, with branches to the subroutines which deliver particular stimulus patterns and then re-adjust the variable used in the MAIN loop to determine occurrence of the next trial, e.g.:

```
$table 5, 10, 7, 6, 8      ' set up inter trial intervals
t1 = 10                    ' set time for first stimulus

MAIN
 if time > t1 then gosub STIM
 goto MAIN

STIM
 trial = trial + 1         ' increment trial counter
 latency = 0
 setTone 3, 200, 64, 100, 2, 300    ' ears, level, event, freq, attack, durat.
 delay until latency > 1   ' wait 1 second
 tone                      ' deliver tone stimulus
 t1 = table(trial)         ' get inter trial interval from table
 return
```

In this program 6 tones will be delivered at intervals of 10, 5, 10, 7, 6 and 8 seconds. The word 'time' in the main loop is a system function which returns time into the recording in seconds, increasing as the recording proceeds. The word 'latency' in the STIM subroutine is a system function which also returns time as the experiment proceeds, but which may be set as well as read, and is used for relative timing purposes, in this case to produce a 1 second delay. Table() is used to get numbers from the list defined by $table. Other words in the program such as 't1' and 'trial' are user variables – names invented by the user to contain temporary values. Any word not listed as a command or function in this manual may be used as a user variable.

Dynamic command summary

BEEP	Make a sound in the computer
BIN8	Operate BIN8 TTL output
CLEAR	*Remove any picture currently showing in the media window
DELAY UNTIL	Wait at this statement until condition is true
EVENT	Read from (x = event) or write to (event = x) PsychLab event port
GOSUB ... RETURN	Jump program lines and return back
GOTO	Jump program lines
IF ... THEN ...	If expression after if is true, do ... after then
INT(..)	Convert fraction to whole number
LATENCY	Increases with time resolved to sampling interval
LOAD [Picture number]	*Load in a picture
NOISE	Operate white noise generator as configured by SETNOISE
PRINT	Write in the debug window
RANDOM	Produce random number between 0 and 255
RELAY	Operate relays on BIN8 device
RETURN	..from GOSUB
SETNOISE	Set up sound noise parameters in sound generator
SETTONE	Set up sound tone parameters in sound generator
SHOCK [level]	Operate shock generator
SHOW	*Show the loaded picture in the media window
STORE	End data acquisition
TABLE(..)	Array function (set up pseudo random conditions)
TIME	Returns absolute time since recording began
TONE	Operate tone generator

*** Commands used for picture presentation.**

BEEP

Rings the computer 'bell'. Duration of sound approximated 0.2 seconds.

BIN8 [value], [event code]

This function sets the TTL output on the BIN8 to the value required, as well as setting the associated event code. The value sent is latched to the TTL sockets and held on the output until a new value and its respective event code is sent. The correct setting must be made in the Options, Set for TTL., menu for this command to work.

[value] (0 to 255) The value to send to the TTL output sockets on the BIN8 binary input/output device.

[Event code] (0 to 255). At the time when the output is latched to the TTL output sockets, the event port is instantly set to the value given for Event code. Note that the event port is designed to permit multiple devices to control it. For this reason, it is advisable to use discrete bit codes for event values, i.e. 1, 2, 4, 8, 16, 32, 64 or 128. If event code 16 for example is asserted to indicate noise, it will only affect that particular bit of the event port. Other bits may be connected to physical TTL sockets on the BIN8 device and simultaneously used for buttons and other input output stimulus cuing functions.

CLEAR

Removes any picture shown in the media window.

DELAY UNTIL ...

A very useful function supported in PsychLab measurement which holds up program execution at this statement until the condition following until becomes true. Basic programmers may understand this command as similar to:

```
 DO
 LOOP UNTIL ...
```

where nothing is placed in the loop. Examples of use:

```
MAIN
 if time = t1 then gosub STIM
 goto MAIN

STIM
 latency = 0
 tone,200
 delay until latency > 1
 if trial = 10 then noise
 trial = trial + 1
 return
```

GOTO [label]

Jump PCC lines until [label] is found. PsychLab will search the PCC file looking for [label], which must be positioned first word on the line and be the only word on that line. The line after [label]

will be the next to be executed. GOTO will be most commonly used to cause a program to jump and repeat certain lines. The difference between GOTO and GOSUB is that after a GOSUB a RETURN is expected. Normal program structure will use GOTO for loops or for branches within a subroutine. GOTO should not be used for jumps out of or back to the MAIN loop - use GOSUB instead.

GOSUB [label]

Jump PCC lines until [label] is found. PsychLab will search the PCC file looking for [label], which must be positioned first word on the line and be the only word on that line, without a space before it. The line after [label] will be the next to be executed. A RETURN instruction will cause program flow to jump back to the line after the GOSUB line. Please refer to example given above for goto. Gosub may be used up to a depth of 6 (ie. there may be up to 6 subroutines within subroutines). Do not use goto MAIN at the end of a subroutine entered by gosub.

IF...then... [expression1] THEN [expression2 or command]

[expression1] is evaluated, and if it is true [expression2 or command] will be interpreted, otherwise operation will continue with the next program line. Often used with a GOSUB command. Example:

```
 if time = t1 then gosub DOSOMETHING
```

This typical example could be used to cause a stimulus at a particular time while an experiment is in progress, or in the case of data analysis, it could cause data to be sent to disk file at a particular time into the recording.

Exactly what will happen when the condition time = t1 is found to be true will be defined by the lines following the subroutine label DOSOMETHING . See GOSUB instruction for more information on subroutines.

To a limited extent, complex expressions may be used for [expression1] or [expression2]:

```
if x + y = (y / z) + 2 then gosub ...
```

If in doubt, do the calculations before the if..then statement.

INT(variable)

Returns the integer (whole number) of a variable (all PsychLab variables are floating point numbers which may have a decimal fraction). Particularly useful when comparing two numbers, or a number is compared with time as floating point numbers may never be exactly equal to each other.

For example:

```
MAIN
 if time = t1 then gosub STIM
 goto MAIN

STIM
 t1 = t1 + random / 10
 t1 = int(t1)
 ...
 return
```

LATENCY

A function that increases with time. Latency may be set (e.g. LATENCY = 0). Latency is useful for close timing of stimuli and analysis time epoches around stimuli. It provides an indication of time resolved to 1mS. In measurement, it is always advisable to use the > comparator rather than = in statements like:

```
if latency > .09
```

instead of

```
if latency = 0.1
```

as there is a risk that the latter condition may never be true owing to the nature of the floating point calculations used by the latency function.

```
MAIN
 if time = t1 then gosub STIM
 goto MAIN

STIM
 latency = 0
 event = 8                          ' switch on warning light
 delay until latency > 1
 latency = 0
 delay until latency > 0.09
 event = 16                         ' operate air puff device
 n1 = n1 + 1
 t1 = t1 + table(n1)
 return
```

LOAD [picture number]

Loads in the picture named in a numbered field of the Set Media Names panel. Note that it can take some time to load each picture, depending on picture size and computer power. Use highly compressed pictures to minimise this delay if rapid successive pictures must be shown. As soon as a picture has been shown, the next one may be loaded, even while the first is still being presented. See also CLEAR and SHOW.

NOISE

Causes the WN noise generator to produce a burst of noise as set up using the SETNOISE command. Very precise timing is achieved with sounds as short as 1mS which may be considered as clicks. See SetNoise for example.

PRINT [item]

Similar to basic print command, but does not support formatting Characters. Only one item is allowed. Only one print statement may be put on a line, if [item] is a string in quotes.

Print displays information in the debug window.

Examples:

```
print variableOne
print "THIS TEXT"
```

RANDOM

Produces pseudo random values between 1 and 255. Values will be whole numbers (integers). Used in the format - e.g.

```
x1 = random
```

Used with $random , random generates completely random Numbers. Without $random , the random sequence may be the same if SAM1 is re-started. To convert the range of RANDOM numbers produced, RANDOM can be multiplied or divided. The int(..) function can be used to convert fractional values to whole numbers.

RELAY [value], [event code]

This function turns on the relevant relay switches on the BIN8 to the value required, as well as setting the associated event code. The relays remain as set until a new value and its respective event code is sent. The correct setting must be made in the Options, Set for TTL., menu for this command to work.

[value] (0 to 15) The binary value to send to control the four(4)relay switches on the BIN8 binary input/output device.
0 = All off (open circuit)
1 = Relay #1 on (makes contact)
2 = Relay #2 on
3 = Relays #1 and #2 on.
.
.
15 = All four relays on.

[Event code] (0 to 255). At the time when the value is sent to switch the relevant relays, the event port is instantly set to the value given for Event code. Note that the event port is designed to permit multiple devices to control it. For this reason, it is advisable to use discrete bit codes for event values, i.e. 1, 2, 4, 8, 16, 32, 64 or 128. If event code 16 for example is asserted to indicate noise, it will only affect that particular bit of the event port. Other bits may be connected to physical TTL sockets on the BIN8 device and simultaneously used for buttons and other input output stimulus cuing functions.

RETURN

Jump back to the next line after the last GOSUB statement.

SETNOISE [ears], [level], [event code], [duration], [revert level]

Sets up the audio generator ready to deliver a noise when the NOISE command is used. The correct setting must be present in the Options, Set for Noise., menu.
Several communications are made between the computer and the audio generator to set it up, each of which takes finite time. In order that the NOISE command can deliver the sound instantaneously, these communications are done with the SETNOISE command prior to the actual noise presentation. Leave at least 10mS delay between the SETNOISE command and the NOISE command to be sure of accurate timing. Note that the DURATION, REVERT LEVEL, and REVERT EVENT CODE parameters are not supported in PsychLab units with version lower than V4.2. If these parameters are not supported, noise duration must be produced by using SETNOISE to set noise volume, wait for at least 10 mS, then use NOISE to turn the noise on, then use SETNOISE to set up a zero volume noise, wait in the program for the required duration, then use NOISE to turn the noise off. PsychLab version number is displayed in the title bar of the PsychLab Properties window.

[Ears] Defines which ear the sound will be heard in:
0 = no ears (off)
1 = left ear
2 = right ear
3 = both ears

[Level] A value between 0 and 255 defining sound volume. Use this setting in conjunction with the manual volume control on the sound generator.

[Event code] (0 to 255). At the time when the noise is presented, the event port is instantly set to the value given for Event code. When the noise goes off, this code is released. Note that the event port is designed to permit multiple devices to control it. For this reason, it is advisable to use discrete bit codes for event values, i.e. 1, 2, 4, 8, 16, 32, 64 or 128. If event code 16 for example is asserted to indicate noise, it will only affect that particular bit of the event port. Other bits may be connected to physical TTL sockets on the BIN8 device and simultaneously used for buttons and other input output stimulus cuing functions.

[Duration] The number of data samples during which the noise should stay on, between 0 and 32767. If sample rate is set to 1000Hz, this value will indicate milliseconds. At other sample rates, the value will need to be calculated (e.g. at 500Hz value 100 will give 200mS, at 100Hz, value 100 will give 1 second). For long or indefinite NOISE presentation, use value 0, which will cause the noise to stay on after the NOISE command until SETNOISE is used to set the level to zero and another NOISE command is used. If PsychLab version is < 4.8, the duration parameter is limited to 0-255, not 0-32767.

[Revert level] optional parameter.
A value between 0 and 255 defining sound volume that will remain after DURATION is completed. This parameter is optional.

```
 'program to test both tone and noise.
 'Duration parameter may only be >255 if PsychLab version is 4.8 or above.

MAIN
 if time > t1 then gosub STIM
 goto MAIN

STIM
 latency = 0
 setTone 3, 200, 64, 100, 2, 300     ' ears, level, event, freq, attack, durat.
 delay until latency > .5            ' allow > 200mS for each set instruction
 tone
```

```
setNoise 3, 200, 128, 40           ' ears, level, event, duration
delay until latency > 3
noise
t1 = t1 + 5
return
```

SETTONE [ears], [level], [event code], [frequency], [attack Decay], [duration]

Sets up the audio generator ready to deliver a tone when the TONE command is used. The correct setting must be present in the Options, Set for Tone., menu.

[Ears] Defines which ear the sound will be heard in:
0 = no ears (off)
1 = left ear
2 = right ear
3 = both ears

[Level] A value between 0 and 255 defining sound volume. Use this setting in conjunction with the manual volume control on the sound generator.

[Event code] (0 to 255). At the time when the tone is presented, the event port is instantly set to the value given for Event code. When the tone goes off, this code is released. Note that the event port is designed to permit multiple devices to control it. For this reason, it is advisable to use discrete bit codes for event values, i.e. 1, 2, 4, 8, 16, 32, 64 or 128. If event code 16 for example is asserted to indicate tone, it will only affect that particular bit of the event port. Other bits may be connected to physical TTL sockets on the BIN8 device and simultaneously used for buttons and other input output stimulus cuing functions.

[Frequency] (0 to 255). Sets frequency, where 0 is about 300Hz, 255 is about 2kHz, and 100 is closely adjusted to be 1kHz.

[Attack/Decay] Sets attach decay where 0 is no delay, attack/decay off, 1 is 50mS and 2 is 100mS.

[Duration] The number of data samples during which the tone should stay on, between 0 and 32767. If sample rate is set to 1000Hz, this value will indicate milliseconds. At other sample rates, the value will need to be calculated (e.g. at 500Hz value 100 will give 200mS, at 100Hz, value 100 will give 1 second). For long or indefinite TONE presentation, use value 0, or omit this parameter, which will cause the tone to stay on after the TONE command until SETTONE is used to set the level to zero and another TONE command is used. This parameter is optional. Versions of PsychLab firmware < 4.8 do not support long Duration parameters, 0-255 is the maximum range instead of 0-32767. the version number is seen in the title bar of the Sam Properties window. Versions below 4.2 do not support this format for SetTone or SetNoise, please apply for a firmware upgrade.

See SetNoise for example of both SetTone and SetNoise commands.

SHOCK [level]

Causes the SHK shock generator to produce a shock. The parameter controls the level of the shock, between 0 (off) and 255 (maximum). Example:

SHOCK 50

presents a shock at level 50. Care should be taken to set the maximum shock level control to prevent accidentally high levels in the case of program malfunction.

SHOW

Shows instantly the last picture that was loaded with a LOAD instruction. See also CLEAR.

STORE

STORE causes data acquisition to cease and prompts the user for a filename to store data. Using STORE to automatically end data acquisition will ensure standard length data files. Example:

```
 if time = 1300 then store
```

TABLE(nn)

Allows reference to a pre-defined table or array of numbers, created using $TABLE . TABLE is a single dimension array of 255 elements (1-255). It is important to note that the first element is numbered 1, not 0, so the variable used to point to the array in the second example below has been preset to 1. Examples:

```
 $table 60,100,170, 220  ' set table elements 1-4
 $table 260,300          ' set table elements 5,6
 n1 = 1
MAIN
 if time = table(n1) then gosub STIMULATE
 goto MAIN
STIMULATE
 n1 = n1 + 1
 .....
 return
```

Values contained in the table may only be whole numbers, between 0 and + 32767. Do not specify positive sign. The first value defined by $table is accessed by TABLE(1) . Up to 255 numbers may be defined. If a particular element in the table does not need to be set to a value, but further elements need to be set, it is not OK to simple put two commas - this will be taken as the last element in the table that is to be set. TABLE(n) may also be used in the format:

```
table(n) = x1
```

However, $TABLE is the most convenient way to put positive values into the table, perhaps analogous to the "DATA" statement in BASIC. If two separate arrays are required, it may be possible to use the one $TABLE array provided in PsychLab but refer to different areas of the table by using two different variables to point to the different sections. See also $TABLE metacommand.

TIME

Returns time into data acquisition in seconds as integer, i.e. whole second values are returned without fractional parts of seconds. TIME is a read only function, it cannot be set, e.g. X = time is OK, but **do not use** *time = X*. TIME is most commonly used for determining inter-trial intervals where relatively long periods are required.

```
MAIN
 if time = t1 then gosub STIM
 goto MAIN

STIM
 ...
 t1 = t1 + 20
 return
```

TONE

Makes a tone on the TG/WN Audio generator, at the level etc. defined in the SetTone command. At the instant the tone is presented, the event level defined in the SetTone command will appear in the event channel.

```
 'program to test both tone and noise.
 'Duration parameter may only be >255 if PsychLab version is 4.8 or above.

MAIN
 if time > t1 then gosub STIM
 goto MAIN

STIM
 latency = 0
 setTone 3, 200, 64, 100, 2, 300    ' ears, level, event, freq, attack, durat.
 delay until latency > .5           ' allow > 200mS for each set instruction
 tone
 setNoise 3, 200, 128, 40           ' ears, level, event, duration
 delay until latency > 3
 noise
 t1 = t1 + 5
 return
```