7 Sampling with multiple states

Introduction

In the *Getting started* chapter you will have encountered the frame state; a value attached to each frame in a Signal data file that can indicate a condition or classification of the frame. You will also have encountered the Enable multiple states checkbox in the General section of the sampling configuration dialog. This chapter describes the uses and control of multiple states in data acquisition.

What can multiple states do?

Simply put, Signal multiple states sampling is a way of generating data files where each frame’s state value is set according to external conditions. The external condition can either be signalled by external equipment or it can be controlled by Signal generating different outputs for each state in use. There are a three different forms of multiple states.

Imagine an experiment involving measuring responses from a test subject to two forms of stimulation. For example, you are recording the EEG evoked response to a tone in either the left ear (stimulus A) or the right ear (stimulus B). You have external equipment to generate the different stimuli and are using Signal to record the result. You want to generate two averaged responses, one for stimulus A and the other for stimulus B.

One way to do this would be to look at signals generated by the external equipment to indicate whether stimulus A or B was used. You could then generate two averages, one for each stimulus. This is what one form of Signal multiple states can do by reading the 1401 digital inputs and using the value read to work out a state code for each frame. This mode of operation is called *External digital* states.

perhaps you go on to make the experiment more complicated. You want four separate stimuli now (maybe two different tones for each ear) and rather than using a fixed stimulus order you want to randomise them. Rather than design hardware complex enough to do this, you can use Signal to generate outputs which control the stimulus, record the outputs used with each frame as the frame state and randomise the state order. The analysis is essentially unchanged apart from now producing four averages, but Signal is now in control of states sequencing and much more complex patterns of stimuli are possible. This mode of operation is called *Static outputs* states.

Finally, rather than just using Signal to select tones from the external equipment, you now want to generate the tones directly from within Signal using the sine wave outputs available from the pulse outputs system. Signal now has multiple sets of pulse outputs, one per state, and as it switches state it switches between the sets of pulses to generate different outputs. This mode of operation is called *Dynamic outputs* states.

These three scenarios briefly describe the three ways that Signal multiple states can be used and touch on some of the ways that states can be sequenced during the experiment. There are many other things that can be done using multiple states, but they all share a common theme; different conditions in the experiment that need to be recorded so that the analysis can distinguish data recorded under the various conditions.

Enabling multiple states

The General page in the sampling configuration dialog contains a checkbox labelled Multiple frame states which is used to enable multiple states in data acquisition. With this checkbox clear, sampling does not use multiple states and all sampled data frames are set to state zero. With this checked, sampling will use multiple states and set the data frames to the appropriate state value.

When multiple frame states are enabled, the sampling configuration dialog gains another page labelled States. This page holds controls used to set up multiple states.

Defining multiple states

The State variation selector at the top left of the States page select the type of multiple states to use from External digital, Static outputs and Dynamic outputs. The controls shown in the dialog change dramatically as each type of multiple states is selected, so the three forms are described separately below.

What type of states do I need?

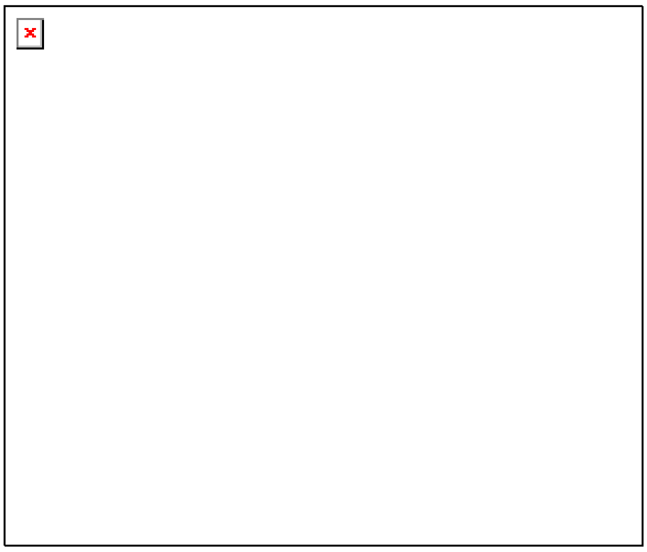
For almost all purposes Dynamic outputs states are the most useful as this allows you to set different output pulses for each state. External digital states can be used only with external hardware that generates digital codes to indicate what is happening, while Static outputs states are limited to unchanging digital and DAC outputs, so both of these are more limited. The main part of this chapter will cover Dynamic outputs states with the other forms of multiple states described afterwards.

Dynamic outputs states

Dynamic outputs states are the most useful and general form of multiple states; each state generates a separate set of output pulses. The actual digital and DAC outputs for each state (along with some other information including the state label) are set by using the Configure Pulses dialog available from the Outputs tab of the sampling configuration, the states page is purely concerned with defining how many states there are and how they are sequenced – how Signal will switch from state to state during sampling.

The Number of extra states item controls the number of extra states, from 1 to 256. Note that this item sets the number of extra states in addition to state zero. Thus in the example shown below there are 5 states possible with codes running from 0 to 4. In many circumstances Signal will use only states 1 to 4, state zero is treated as an background or idle state. This also controls the states available in the pulses configuration dialog; you should set the extra states you want here before you set up the pulse outputs themselves.

State sequencing



Because in Dynamic outputs mode Signal is controlling the states (as it is in Static outputs mode), we need ways of defining which state is used when. The simplest way to do this is to control the state manually (which can be done using a control bar), but often we want some form of automatic states sequencing. Signal provides essentially two separate forms of states sequencing – numeric or protocol. In the numeric modes (there are three of these) simple numeric or randomised sequencing is provided while, in protocol mode a precisely defined complex sequence of states can be generated.

Numeric (non-protocol) sequencing modes

In non-protocol states sequencing modes, the user is limited to specifying how many of each state are to be used and how they are to be randomised (if at all).

The Individual repeats checkbox selects if each state should be repeated a different number of times or if all states are repeated the same number of times. If it is clear, the Repeats item is used to set how many times each state is repeated. If the checkbox is set, separate controls are used to set the repeats wanted for each state (see below).

The Cycles before idle item sets how many times the full set of states (states * repeats) is repeated before states sequencing is automatically stopped and Signal switches automatically to state 0. Set this item to the number of sequencing cycles you want or to 0 if you want states sequencing to repeat forever until stopped.

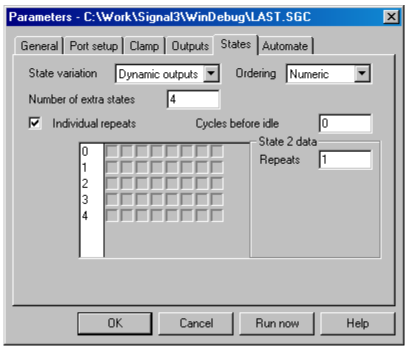
The Ordering selector to the top right of the states page selects the type of sequencing to be used, this can be set to:

Numeric	In this mode the extra states are used in numerical order with each state being used a specified number of times (as set by the overall or individual repeat counts). First state 1 is used n times, then state 2 and so forth. Once the last state has been done, one sequencing cycle has been completed and the sequencing either stops or restarts with state 1 as set by the Cycles before idle control.
Random	In this mode, one cycle of the sequencing again uses each state the number of times specified by the repeats, but the order of the states is randomised. So, with two states and two repeats, each cycle lasts for four frames, two of which are certain to be state 1 and two state 2, but in a random order. When the sequencing starts another cycle, the states order is re-randomised.
Semi-random	This is a slight modification of Random mode, where the repeats are not all randomised together across a cycle. If there are n states then the first n frames will contain one of each state (in a random order), as will the next n and so forth. So, with 2 extra states and two repeats, the first two frames will contain one each of states 1 and 2, in random order, and the next two frames will again contain one each of state 1 and 2, again randomised. So really this achieves the same thing as setting the number of repeats to 1, but a sequencing cycle still consists of each state being used the number of times specified by the repeats.
Protocol	In this mode, a separate protocol dialog is used to define arbitrary sequences of states repeated an arbitrary number of times. This, of course, is not a numeric sequencing mode and is documented separately, see the section on Protocols below. When Protocol mode is selected, all controls for state repeats and cycles are hidden and replaced by a Protocols… button which provides the protocol dialog.

When sampling using dynamic outputs Signal provides controls for the state and states sequencing, see the section *Controlling multiple states online* below.

Individual state repeat counts

When the Individual repeats checkbox is set, the Repeats item that sets the repeat count for all states is hidden. Instead the dialog shows a state selector and repeat so you can set repeats for each state. The state selector is derived from the digital output control used in other modes, to set the repeat count for a state, click on the desired state and edit the Repeats value shown in the state data box.

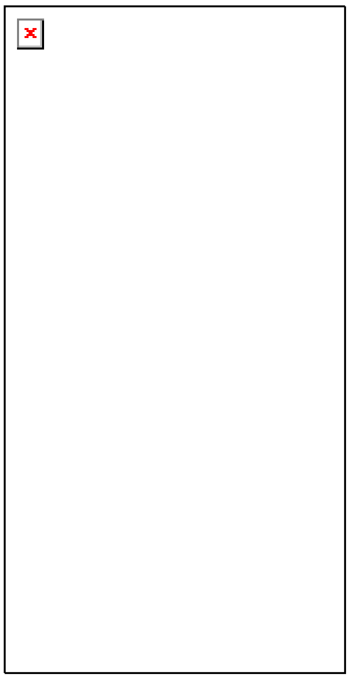

Parameters - C:\Work\Signal3\WinDebug\LAST.SGC

State variation: Dynamic outputs
Ordering: Numeric
Number of extra states: 4
[✓] Individual repeats
Cycles before idle: 0
State 2 data
Repeats: 1

Individual repeat counts are used in much the same way as an overall repeat value, only each state has a separate count. In Numeric ordering each state is repeated in turn the specified number of times within each sequencing cycle, whereas with Randomised ordering each state is repeated the set number of times, but the order of states within each sequencing cycle is randomised. Semi-random ordering does not work well with individual repeats.

State sequencing by protocol

A protocol consists of a list of up to 10 steps, each step setting the state that will be used, a repeat count and a step to use next. A protocol can use up to ten steps, protocols can loop and be chained together to produce longer sequences of states. Up to 50 separate protocols can be defined within a single sampling configuration, giving great flexibility.



Protocols are defined using the protocols dialog which is obtained by pressing the Protocol… button on the states page. The dialog contains a selector at the top that is used to select a protocol for viewing and editing. To create a new protocol press the Add Protocol button, while protocols can be deleted using the Delete button.

The protocol name can be changed by directly editing it in the protocol selector. Blank protocol names are not allowed and will be rejected.

The checkboxes at the top of the protocol details set general options and what happens at the start of protocol execution. The Create toolbar button for protocol item enables a separate button for this protocol in the states control bar – see the section *Controlling multiple states online,* below. Cycle protocol states only after write controls the sequencing of protocol steps; if it is checked then the protocol sequence does not advance unless data file frames are written to disk, if the data is not written or a sweep is rejected then the protocol does not advance. Turn on writing at protocol start causes writing of sampled sweeps to disk (normally controlled by the Write to disk at sweep end checkbox in the sampling control panel) to be automatically turned on when the protocol starts. Reset pulse steps at protocol start causes all varying pulses defined in the pulses dialog to be reset to their initial state when the protocol starts.

The table below defines the protocol steps. There are ten steps in a protocol, each one specifies a State, Repeats and a Next step. These specify the state to be used, the number of times to repeat this state and the step to go to (from 1 to 10) when this step is done. A step is marked as being the end of the protocol by setting it's next step to zero.

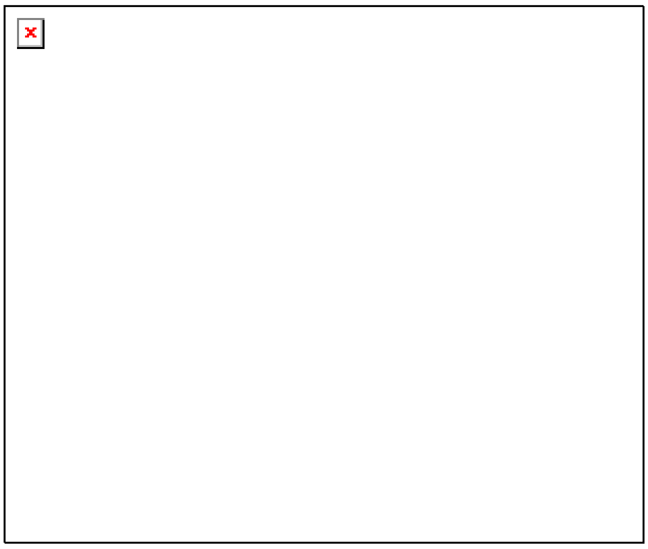
When protocol execution begins it starts with step 1. The state set by the State field in step 1 is set and is used the number of times set in the Repeats field. Following this the protocol switches to the step number set by the Next field. This process continues until it is ended by encountering a Next item of zero. In the example shown above. step 1 repeats state 1 four times and then goes to step 2. This repeats state 3 eight times, after which the protocol ends. If the Next item for step 2 were set to 1, the protocol would run forever or it could be set to 3 for a more complex sequence.

The Repeat count for entire protocol item below the step table controls the number of times that the protocol repeats before it ends. Set this to 1 for a protocol that goes through the steps only once (as in the example above) or set it to a larger number for a protocol that repeats a set number of times before ending. For example, if you set the repeat count to three in the example above, the protocol would run for 36 frames before ending. If you set this item to 0 the protocol will repeat forever until stopped.

The items below the repeat count control what happens when protocol execution ends. The At end selector selects between Finish and a protocol that will be ‘chained-to’, chaining to a protocol allows more complex sequences than is possible with just ten steps. If the At end selector is set to Finish protocol execution finishes, otherwise execution switches to the protocol selected and carries on. When a protocol is chained-to, it starts off completely normally so that the Turn on writing and Reset pulse steps checkboxes take effect. These checkboxes do not take effect if the last step in the protocol has a Next item of step 1, so it is meaningful to allow a protocol to chain to itself..

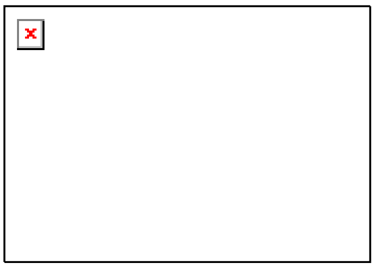
The checkboxes at the bottom control what happens when a protocol actually finishes and are disabled if the protocol chains to another protocol. State zero when protocol finishes enables automatic switching to state zero at the end; if it is clear then the last state set by the protocol will continue to be used. Turn off writing when protocol finishes controls disabling of writing sampled data to disc.

External digital states



External digital states are the simplest but most limited form of multiple states. External equipment generates a digital code of up to 8 bits corresponding to the current experiment state. Signal reads this code from the 1401 digital inputs 8-15 at the end of a sampling sweep and uses it to set the state for each sampled data frame. There is no software control over the state values, so all of the state sequencing control are not present and no difference in Signal’s behaviour for the different states.

The number of extra states is set as normal, while the rest of the states page is used to defines the mapping from digital input values to the frame state. The Digital inputs enable checkboxes at the bottom of the dialog control which inputs are used for this; unchecked inputs are ignored.



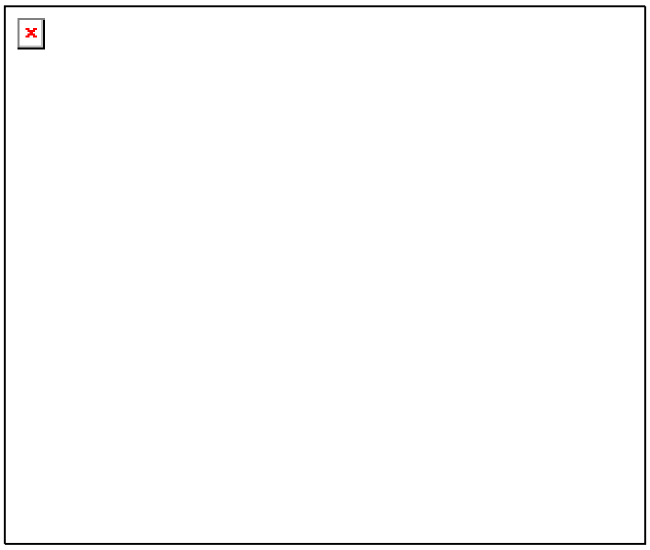
The main area in the centre of the dialog defines, for the enabled digital input bits only, the bit patterns that correspond to the various states. The bits 0 to 7 are shown in a row, with 0 on the left. A blank location corresponds to a zero bit (low or 0 volts), a checked location gives a set bit (high or 5 volts). When Signal is sampling, it reads the digital inputs at the end of each sampling sweep. The bits read are then checked against the bits for each of the states starting with state 1 and the frame state is set to the first one that matches. If no match is found then the frame state is set to zero. The bits for state zero are ignored. See the Sampling data chapter for details of the digital input connections.

The buttons to the left of the digital bit controls set or modify the state bit patterns to various useful ways. These are intended to allow simple patterns to be set up quickly and to help to produce more complex ones:

Invert	This inverts all of the bits for all states. This is useful for converting all zeroes to all ones and a walking one into a walking zero.
0000	This sets all of the bits for all states to zero. This is useful for clearing the control before entering new patterns.
0100	This sets the bits to zero with a walking 1. This leaves state zero all clear, sets bit 0 for state 1, bit 1 for state 2 and so forth. The pattern is not adjusted to skip disabled digital input bits.
1234	This sets the bit values to count the states, using binary code. Thus state 1 has just bit 0 set, state 2 has bit 1 only and state 3 has both bits 0 and 1. Again, the pattern is not adjusted to skip disabled digital input bits.

When sampling using External digital states, Signal behaves much as it does with multiple states disabled. The only difference is that the state value for sampled data frames varies according to the digital inputs.

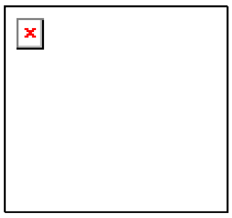
Static outputs states



Static outputs states are more complicated than External digital states as Signal controls the state of each sampled data frame. At the start of a sampling sweep, Signal writes values to the 1401 DACs and digital outputs according to the current state of the experiment. These outputs could select the stimulus or have other effects as required.

In addition to defining the values to be output, Static outputs states requires that you also set up how Signal will sequence the states. This adds quite a number of extra controls to the dialog, for details of these see the section on *States sequencing* above.

The main box in the centre of the dialog now defines the values to be written to the digital outputs for each sweep. Because these are now digital outputs, the enables for the bits to be used are in the Outputs page, but otherwise this and the adjacent buttons behave in the same way as for External digital mode. The actual digital outputs and connector pins used are the same as for pulse outputs (see the *Pulse outputs during sampling* chapter).



The State data box to the right of digital output bits sets the DAC outputs for a selected state. You can select the state for which values are shown by clicking on the digital outputs line for that state. The DAC outputs used (0 to 3 only are available) are enabled and disabled in the Outputs page of the configuration.

Static output states can only be used with the outputs type set to None in the Outputs page, as otherwise the state values would conflict with the other outputs. When sampling using Static output states, controls for the state and states sequencing are now provided, see the section *Controlling multiple states online* below.

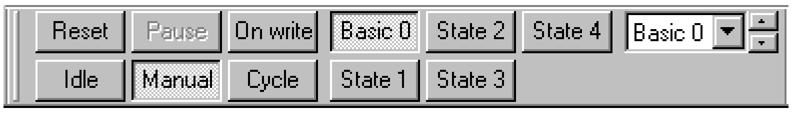
Controlling multiple states online

When sampling using dynamic outputs (or static) multiple states, Signal provides a states control bar to allow you to control the sampling state and sequencing. The control bar can be docked at the edges of the Signal application or be left floating. When sampling starts the states control bar will be shown initially and can be hidden or shown during sampling by using the sample menu or a popup menu that is displayed if you right-click on an unused part of the toolbar area.

The states control bar is set up differently depending upon whether numeric/random or protocol sequencing are being used; without protocols extra buttons for controlling states sequencing are provided whilst when using protocols extra items to select and run a protocol are provided along with optional individual protocol buttons.

Non-protocol ordering

The states control bar contains a number of buttons and controls:

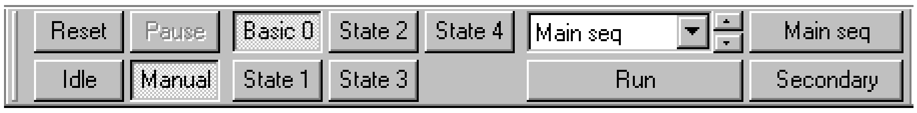


Reset	Press this button to reset any states sequencing in operation (so that the state sequence restarts at the beginning) and also to reset any varying pulses to their initial state.
Idle	Press this button to force sampling and states sequencing to idle. It switches manual control of states on, sets state zero and turns off writing to disk.
Pause	Press this button to pause any automatic state sequencing that is in progress. This option does not pause the sampling itself which continues to run using the current state (use the sampling control panel if you want to pause the sampling). While states sequencing is paused this button is shown depressed; press it again to resume sequencing.
Manual	Press this button to terminate any automatic sequencing in progress and to switch to manual control of states. With manual control, the frame state is controlled interactively by the user, sampling always begins with manual control selected.
On write	Press this button for automatic states sequencing with the states changing only if the previous data was written. The states sequencer will have control of the frame states and will move on to the next stage after a sampling sweep only if the sweep was saved to disk. Rejected sweeps will repeat the same state until the data is accepted. This allows artefact rejection and interactive sweep accept/rejection without missing out states from the sequence.
Cycle	Press this button to start automatic states sequencing with the states always changing. The state sequencer will have control of the frame states and the sequencer will always move on to the next stage after a sampling sweep, regardless of whether the data is written to disk or not.
Basic 0 …	Press these buttons when in manual mode to use a state immediately. Buttons for unused states are hidden, as are buttons for states numbers greater than 9. If you have set labels for the states in the pulses configuration dialog, these labels will be used in the buttons instead of the standard names. During automatic states sequencing these buttons are disabled but they are pressed automatically to show the state in use.
State n	To the right of the individual state buttons is a state selector and spinner that can be used when in manual mode to choose any state from those available. The selector is most useful when there are more than 9 states, the limit for individual state buttons. As for the buttons, if you have set a label for a state

this is used instead of the simple state name. Selecting a state uses it immediately. During automatic states sequencing the current state is shown.

Protocol ordering

When protocols are used, some of the states bar controls are hidden and other controls are now displayed. Those controls that are retained behave in the same way as described above, while the On write and Cycle buttons are hidden because the individual protocols select between these two modes of operation. If the number of states is less than ten, so that they can all be represented by the individual state buttons, the main state selector is also hidden to save space. The extra items provided are the protocol selector and protocol run button and the individual protocol buttons:



Protocol	To the right of the state buttons (or the state selector if it is visible) is a protocol selector and spinner that can be used to select a protocol from those available. Unlike the state selector, selecting a protocol does not execute the protocol. During execution of a protocol the current protocol in use is shown so you can see what is going on if you use chained protocols. If all of the protocols have individual buttons then the protocol selector is not shown.
Run	This button is shown below the protocol selector. Pressing it causes Signal to switch out of manual mode and start executing the selected protocol immediately, terminating any other executing protocol. Like the protocol selector, if all protocols have individual buttons this button is not shown.
Buttons	These are buttons created for the individual protocols which are arranged to the right of (or below) the protocol selector and are labelled with the protocol name. A protocol button will be created if the Create toolbar button for protocol checkbox in the protocol definition is set and there are not too many protocol buttons – the limit is 20. Pressing one of these buttons causes the relevant protocol to be executed immediately.

Starting a protocol

Execution of a protocol is begun by the user pressing the Run button with that protocol selected (or the button for an individual protocol), or by another protocol chaining to it or by means of the script language.

Stopping a protocol

Execution of a protocol is stopped when the protocol finishes executing, by the user pressing the Idle or Manual buttons in the states control bar or by another protocol being started by whatever means.

Auxiliary state hardware

In addition to the standard multiple states facilities that control 1401 outputs, it is possible to install support for auxiliary states hardware with Signal. Auxiliary states hardware is separate, external hardware (most often a type of stimulator that cannot be adequately controlled using the 1401 outputs) that is set up in a different way for each state.

Auxiliary states hardware support is provided by means of a DLL that is copied to the Signal installation directory; a separate DLL is normally supplied for each type of hardware. When auxiliary states hardware is installed an extra button (labelled with the external hardware name) is provided in the States page to allow the settings for the hardware for each state to be defined. In addition, the `SampleAuxStateParam()` and `SampleAuxStateValue()` script language functions can be used to read and write the settings.

See the appendices for details of the individual auxiliary states hardware supported.