Logical Switch Tutorial 1
Introduction

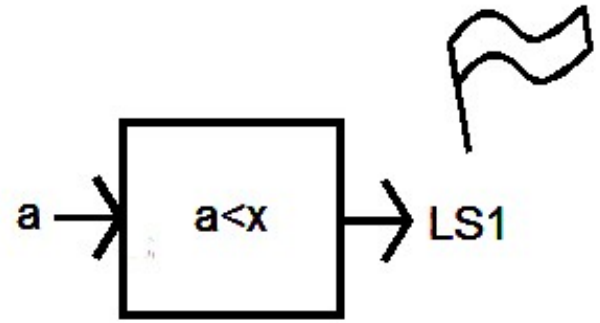
Manually activated Switches are labeled SA to SH: SF is a two position switch, SH is a two position momentary switch and the others are three position switches. Each of these switches provide signals to the computer. The computer checks each switch position regularly (30 times per second).

Example:

Switch SA can be in one of three positions. The computer reads +100, 0, and -100. Each position tells the computer something different. You can use the switch values as input to functions or you can use the values just as "flags" to tell what position the switch is in. The computer decides what to do depending on what you have selected that switch position to do when you set up your model. Maybe you set SA↑ to select Flight Mode 2. When the computer sees that you have positioned SA to SA↑ it will set Flight Mode 2.

Logical Switch Tutorial 1 - p2

Logical Switches are like manual switches except that they can be activated automatically, in response to some condition sensed by the OpenTX program. The computer can be set up to watch for some condition to happen, then “raise a flag” to signal that the condition has been seen. The computer checks the state of the Logical Switch “flags” along with the manual switch states as it prepares to assemble the next set of commands to send to the receiver in your model.

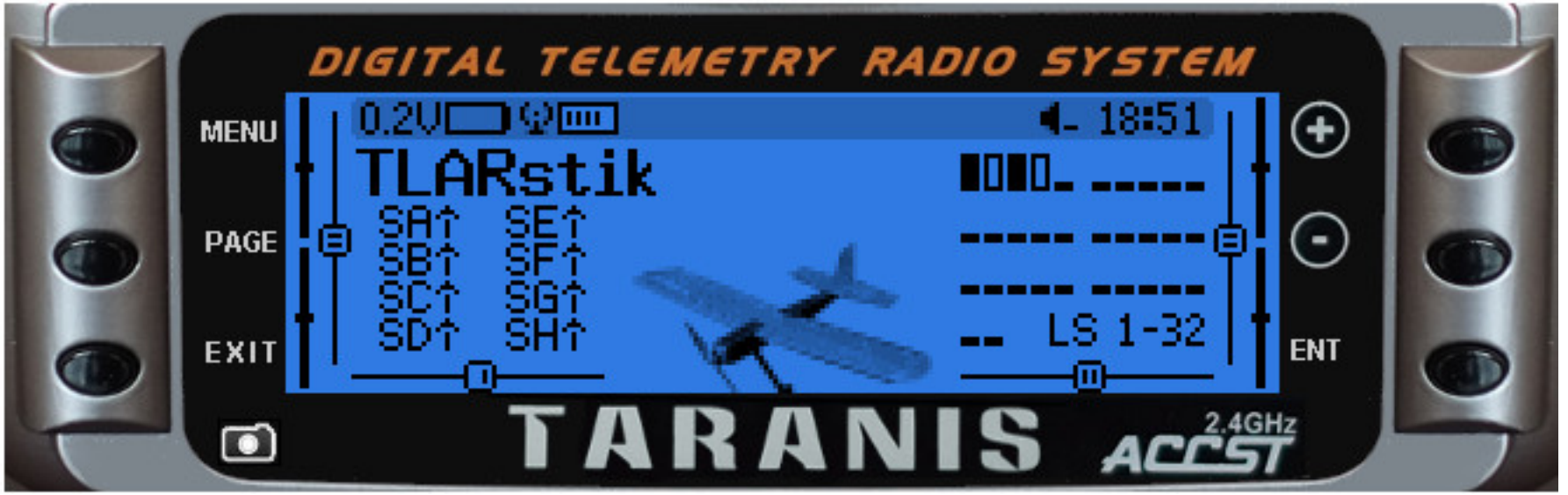


- a = Value being observed
- a<x tells what the “Black Box” function looks for
- LS1 is the output of the Black box if a<x is found to be true

If (a = Throttle value) and (x=-95),
then when the throttle moves to output a value less than -95, LS1 will signal (flag)
Otherwise, LS1 will not signal.

A Logic Switch can be used to sense many conditions, and the resulting signal can command any of a large number of changes to the model setup

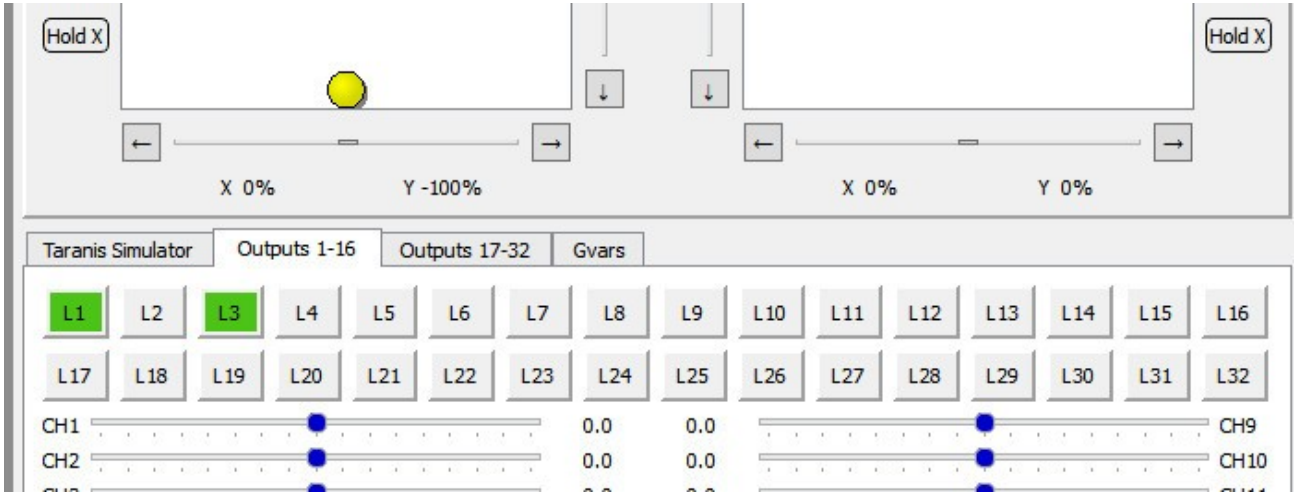
Logical Switch Tutorial 1 – p3
Taranis Logical Switch Screen



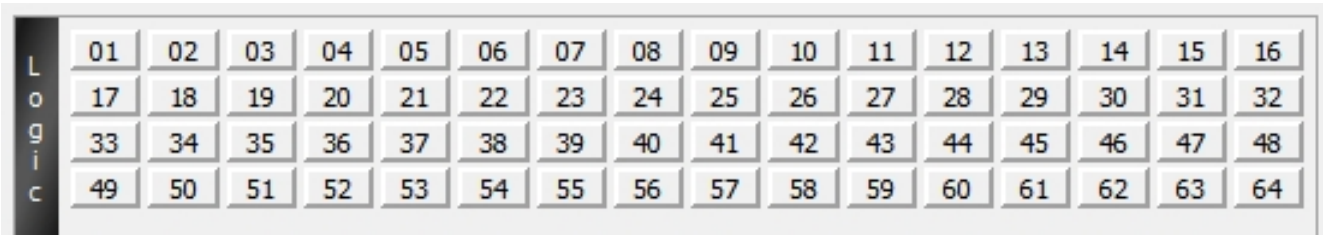
The Logical Switch screen shows the state of the Logical Switches (LS)
The square boxes show which LS has been programmed
Black square boxes show which LS has been set TRUE and so is active.

Logical Switch Tutorial 1 – p4
Companion Logical Switch Screen

2.1.9

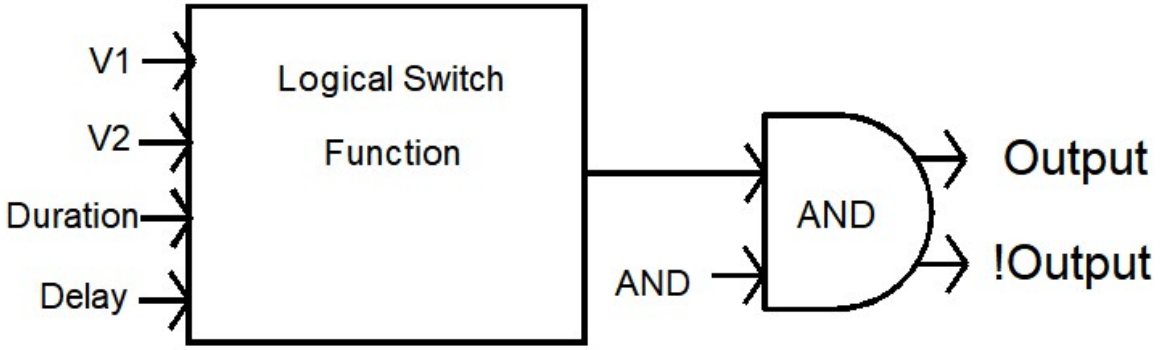


2.2.2



The Logical Switch screen shows the state of the Logical Switches (LS)
The square boxes show all LS states, whether programmed or not
Green square boxes show which LS has been set TRUE and therefore are active

Logical Switch Tutorial 1 - p5



Function	Select Logical Switch (LS) Function
V1	Input Signal Source 1
V2	Input Signal Source 2
And Switch	Input Signal to the AND Switch
Duration	Time the LS Function is TRUE after it is triggered
Delay	Time before the LS Function is triggered after it is signaled

Logical Switch Tutorial 1 – p6
Switch Functions

a = x When Input V1 is equal to the Value in V2

a ~ x When Input V1 is almost equal (+/- 0.9) to the Value in V2

a < x When Input V1 is less than the Value in V2

a > x When Input V1 is greater than the Value in V2

These four functions look at input value V1 and compare it to a specific value of V2 which you define. The input V1 is selected from a list of a drop-down menu. The dropdown list includes discrete (switch and Logical Switch) and analogue (controls, variable knobs and sliders) sources of input.

Even comparing a ~ x may be difficult because the value must be about +/- 0.9 from the set value, so approximate is still very close.

Logical Switch Tutorial 1 – p7
Switch Functions

a = b When Input V1 is equal to input V2

a ~ b When Input V1 is almost equal (+/- 0.9) to input V2

a < b When Input V1 is less than input V2

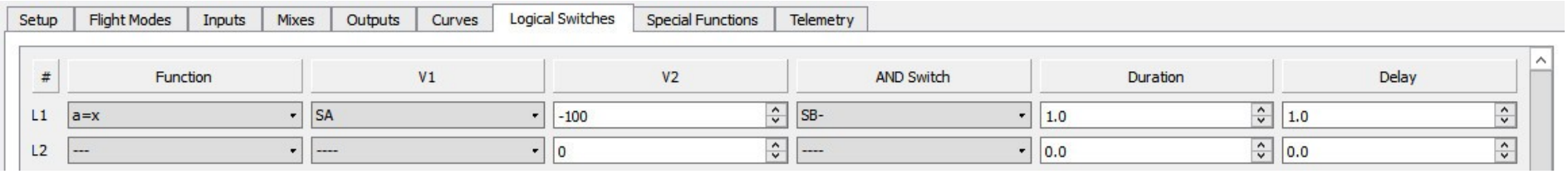
a > b When Input V1 is greater than input V2

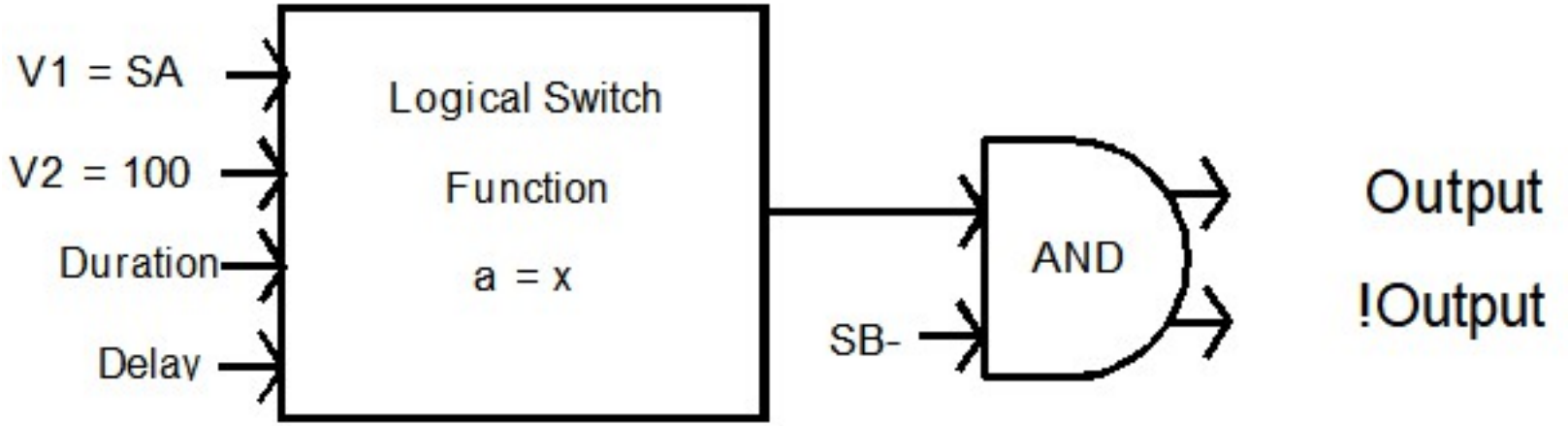
These functions compare the value of an input V1 against the value of a second input V2. Each input is selected from a list of a drop-down menu.

Comparing a = b and a ~ b may be very difficult when you are comparing two analogue signals such as Left Slider and Potentiometer S1 because it may be difficult to set each one exactly on the same value.

Even comparing a ~ b may be difficult because the value must be about +/- 0.9 from the set value, so approximate is still very close.

Logical Switch Tutorial 1 - p8





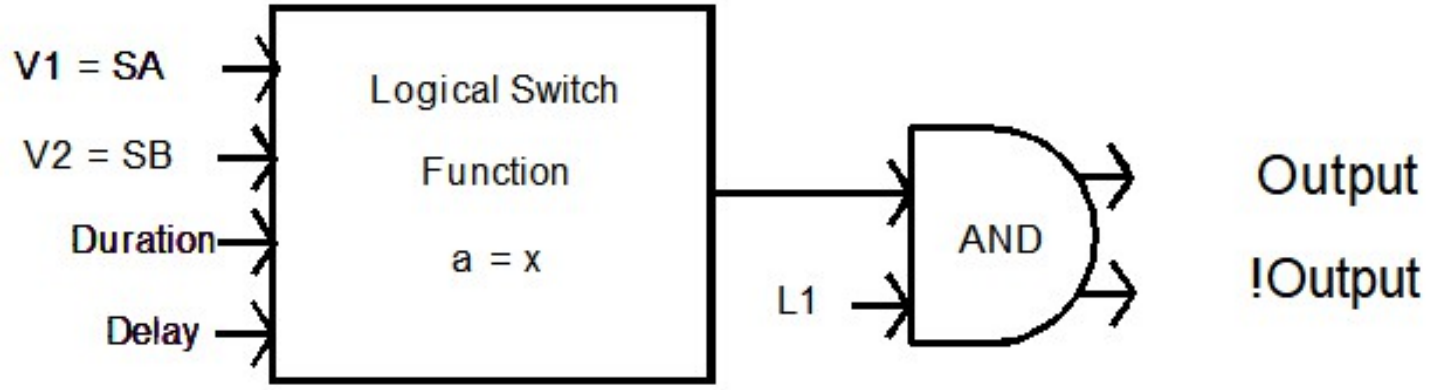
Function	a = x
V1	Manual Switch SA
V2	Value = -100
And Switch	Manual Switch SB-
Duration	1.0 sec
Delay	1.0 sec

V1 is compared to v2 (-100)
If Switch A reads -100
Then the input to the AND Gate will be TRUE for 1 second after a delay of 1 sec

AND if Switch SB is set in the Middle (SB-) the LS output will be TRUE

Logical Switch Tutorial 1 - p9

#	Function	V1	V2	AND Switch	Duration	Delay
L1	a<x	[I3]Thr	-95	----	0.0	0.0
L2	a=b	SA	SB	L1	0.0	0.0



Function	a = b
V1	Manual Switch SA
V2	Manual Switch SB
And Switch	Logical Switch L1
Duration	Minimum Duration
Delay	No delay

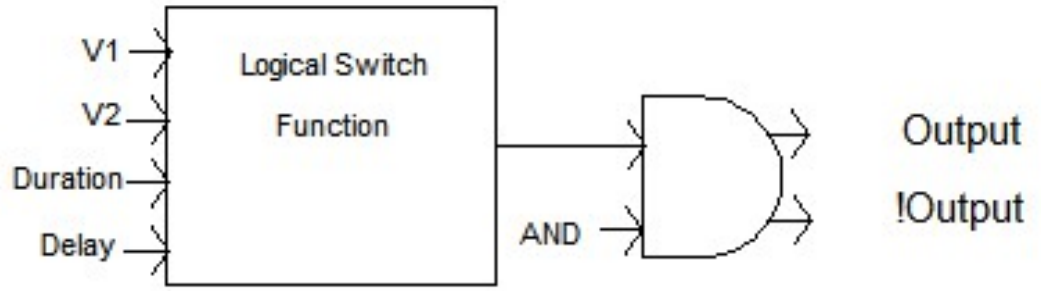
L1 will be true when the Throttle Is moved to lower than -95

L2 will be TRUE when both SA and SB are set the same AND L1 is TRUE

So L2 output cannot be TRUE if the Throttle is above -95

Logical Switch Tutorial 1 – p10
Exercise 1

#	Function	V1	V2	AND Switch	Duration	Delay
L1	a=x	SA	100	----	0.0	0.0
L2	a=x	SA	100	----	2.0	1.0
L3	a=x	SA	100	----	0.0	1.0
L4	a=x	SA	100	L3	0.0	1.0
L5	---	----	0	----	0.0	0.0



As an exercise, set up logical switches as above.

Activate Manual Switch SA.

Observe the results on the Taranis Display and the Companion simulator

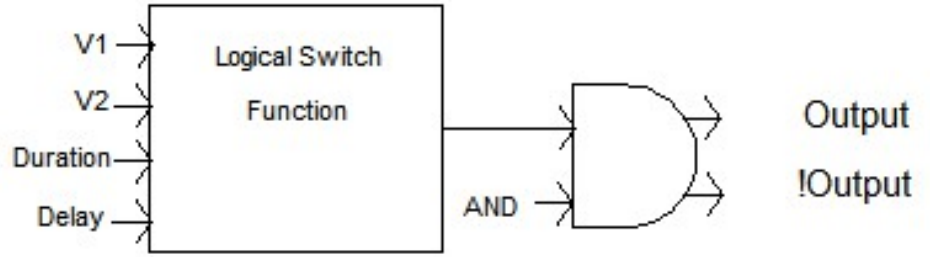
Observe the result on both the Output and the !Output

Observe the AND Switch and its control of both Outputs

Observe the result when a value is included for Duration, Delay, and both together

Logical Switch Tutorial 1 – p11
Exercise 2 – 5 Position Rotary Switch

#	Function	V1	V2	AND Switch	Duration	Delay
L1	a<x	S1	-85	----	0.0	0.0
L2	a<x	S1	-45	!L1	0.0	0.0
L3	|a|<x	S1	45	----	0.0	0.0
L4	a>x	S1	45	!L5	0.0	0.0
L5	a>x	S1	85	----	0.0	0.0
L6	---	----	0	----	0.0	0.0



As an exercise, set up logical switches as above.

Turn Rotary Switch S1

Observe the results on the Taranis Display and the Companion simulator

See 5 individual flags as you rotate S1. Could you use this functionality?

Note:
!L1 means NOT L1 or the opposite of L1. !L1 is True when L1 is False
a<x means a less than x
a>x means a greater than x
|a| means V2 is evaluated as positive whether V2 is positive or negative

Logical Switch Tutorial 1 – p12
Exercise 3 – 18 Logical Switches from 3 manual switches

#	Function	V1	V2	AND Switch	
L1	AND	SA↑	SB↑	SC↑	0.0
L2	AND	SA↑	SB-	SC↑	0.0
L3	AND	SA↑	SB↓	SC↑	0.0
L4	AND	SA-	SB↑	SC↑	0.0
L5	AND	SA-	SB-	SC↑	0.0
L6	AND	SA-	SB↓	SC↑	0.0
L7	AND	SA↓	SB↑	SC↑	0.0
L8	AND	SA↓	SB-	SC↑	0.0
L9	AND	SA↓	SB↓	SC↑	0.0
L10	AND	SA↑	SB↑	SC↓	0.0
L11	AND	SA↑	SB-	SC↓	0.0
L12	AND	SA↑	SB↓	SC↓	0.0
L13	AND	SA-	SB↑	SC↓	0.0
L14	AND	SA-	SB-	SC↓	0.0
L15	AND	SA-	SB↓	SC↓	0.0
L16	AND	SA↓	SB↑	SC↓	0.0
L17	AND	SA↓	SB-	SC↓	0.0
L18	AND	SA↓	SB↓	SC↓	0.0

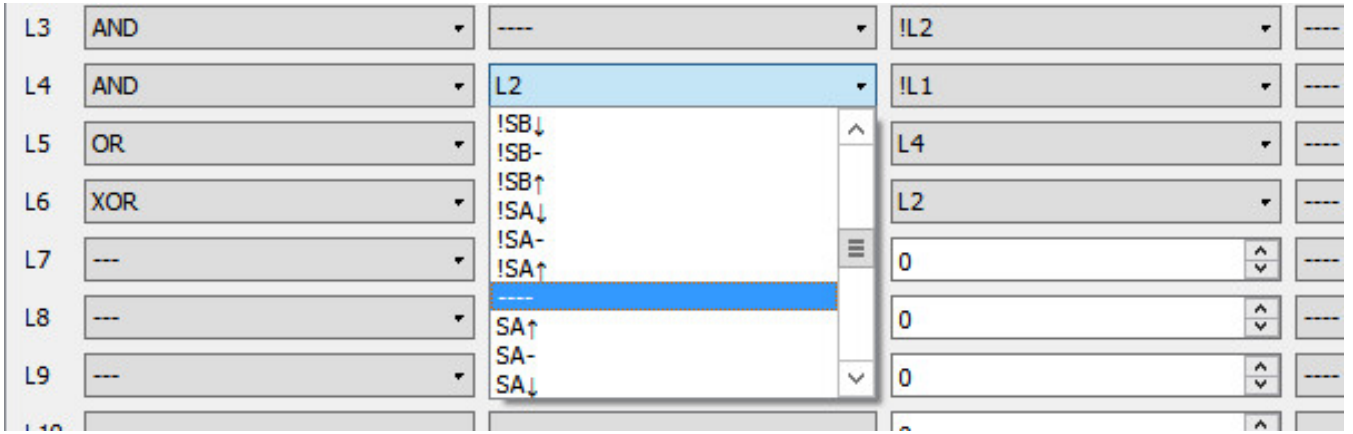
Logical Switch Tutorial 1 – p13
Exercise 3 – continued

Two three position switches (SA and SB) are used to select nine unique states.

The third three position switch (SC) is used to activate the Logical Switches formed by the AND of the SA and SB switches. This will activate 18 Logical Switches

The middle position (SC--) of switch SC provides a neutral state so you can select the Logical Switch you want before you activate it with SC.

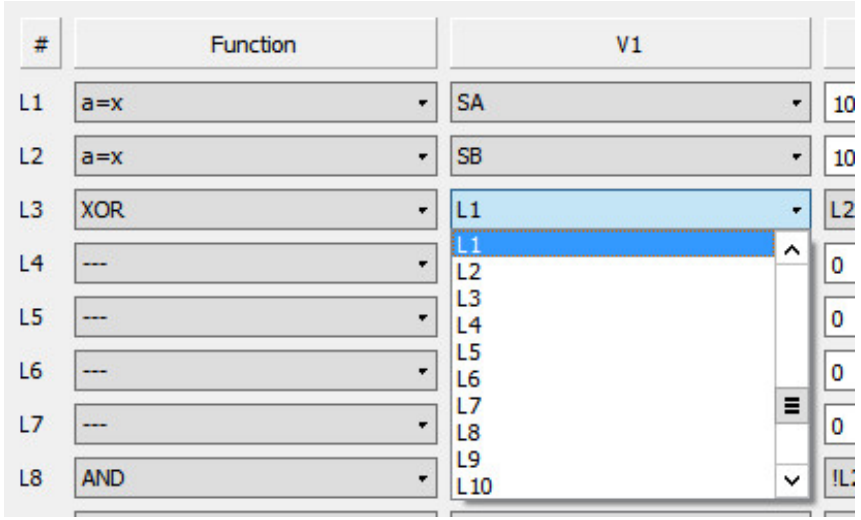
Logical Switch Tutorial 2 – p1
Logic Gates – AND, OR, XOR
Initial Considerations for manual switches

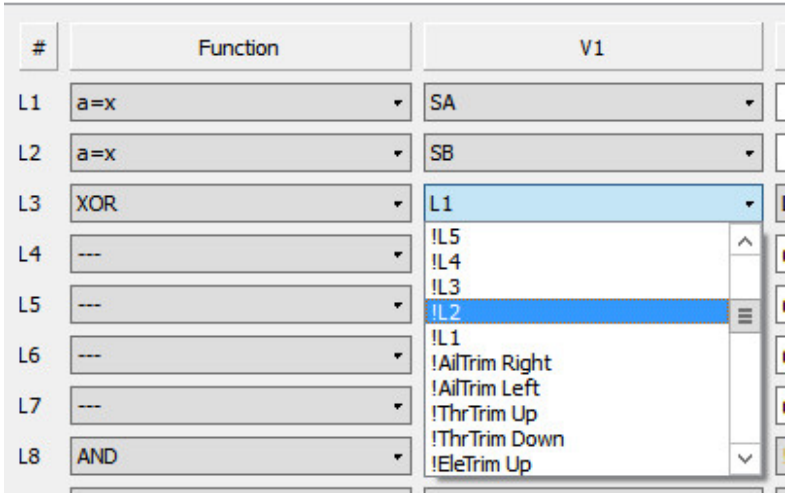


Gate Inputs:

- AND, OR and XOR Gate inputs need to be either TRUE (100) or FALSE (-100). They cannot be analogue (anything from 100 to -100)
- Each manual switch has two or three positions. Each position can be either TRUE (eg. SA↑ can be TRUE or FALSE) or FALSE (!SA↑ can be TRUE or FALSE) So a three position switch can output six possible states.
- You can select SA↑ or !SA↑ as an input to a Logic Switch but you must be very aware of what value of SA↑ you are expecting.
- When you select SA↑ as an input you want SA↑ to be selected active.
- When you select !SA↑ as an input you want SA↑ to be deselected.
- When !L2 is desired as an input to the AND function the L2 signal will be FALSE.

Logical Switch Tutorial 2 – p2
Logic Gates – AND, OR, XOR
Initial Considerations for Logical Switches





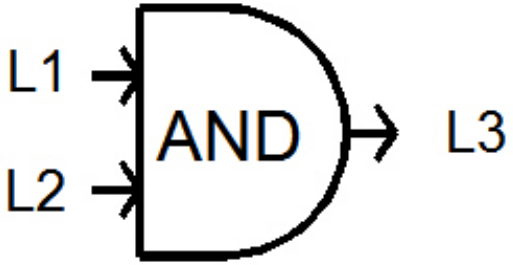
Gate Inputs:

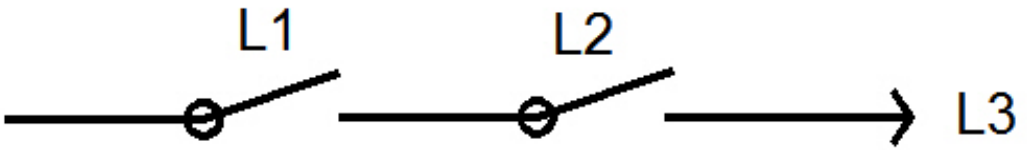
- AND, OR and XOR Gate inputs need to be either TRUE (100) or FALSE (-100). They cannot be analogue (anything from 100 to -100)
- Outputs from Logical Switches can be only 100 and -100. TRUE is 100, FALSE is -100.
- BUT: You have a choice. When L1 is TRUE, its opposite !L1 is FALSE. When L1 is FALSE !L1 will be TRUE. So you can choose to use either L1 or !L1 depending on the logic you are needing.

The following descriptions will show generic Truth Tables indicating TRUE (T) and FALSE (F). A closed switch will indicate TRUE and an open switch will indicate FALSE.

Logical Switch Tutorial 2 – p3
Logic of AND

#	Function	V1	V2	
L1	a=x	SA	0	----
L2	a=x	SB	-100	----
L3	AND	L1	L2	----





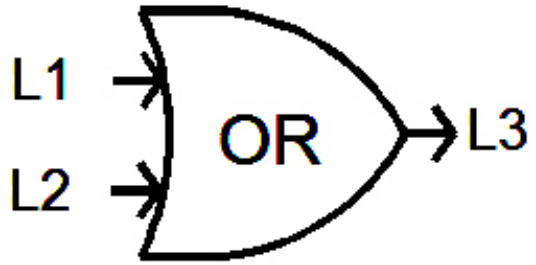
AND

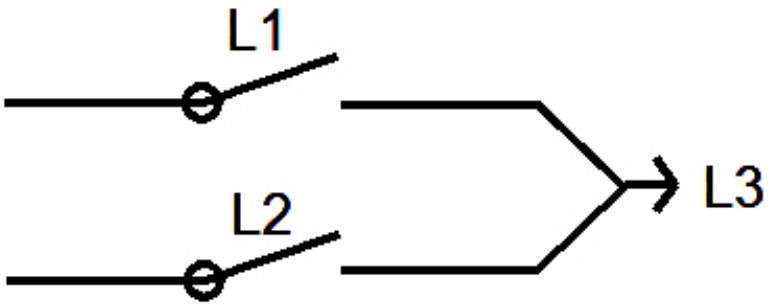
L1	L2	L3
F	F	F
F	T	F
T	F	F
T	T	T

Both inputs must be TRUE for AND to signal the AND Logical Switch Output TRUE

Logical Switch Tutorial 2 – p4
Logic of OR

#	Function	V1	V2	AND
L1	a=x	SA	0	----
L2	a=x	SB	-100	----
L3	OR	L1	L2	----





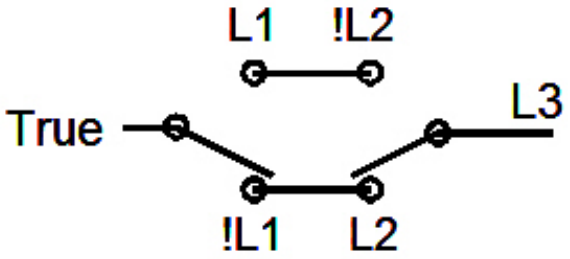
OR

L1	L2	L3
F	F	F
F	T	T
T	F	T
T	T	T

At least one input must be TRUE for OR
to signal the OR Logical Switch Output TRUE

Logical Switch Tutorial 2 – p5
Logic of XOR

#	Function	V1	V2	
L1	a=x	SA	100	----
L2	a=x	SB	100	----
L3	XOR	L1	L2	----



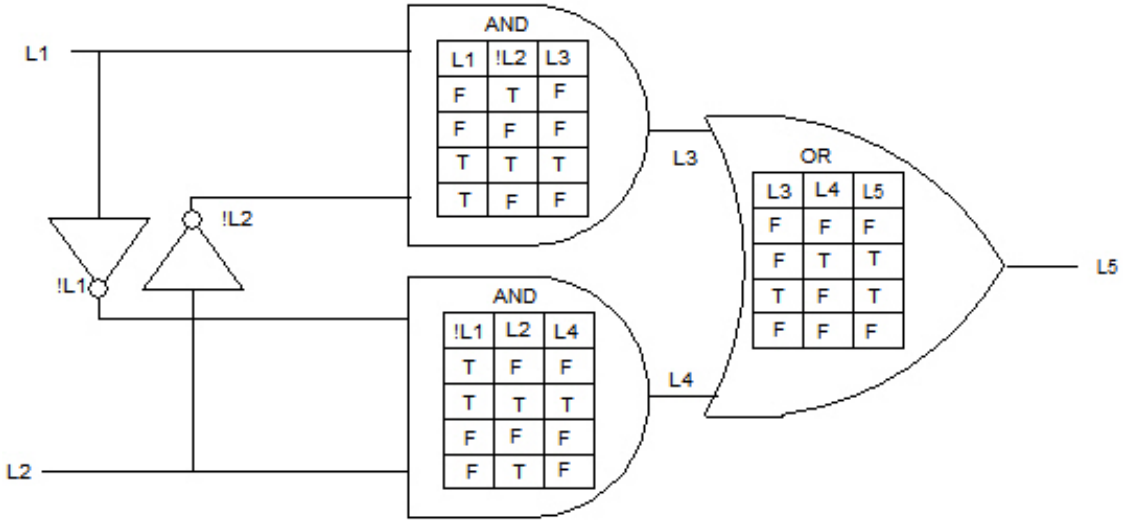
XOR

L1	L2	L3
F	F	F
F	T	T
T	F	T
T	T	F

- The Output of L1 will be TRUE if the Switch A Output Is 100.
- The Output of L2 will be TRUE if the Switch B Output Is 100.
- Either V1 (The output of L1) or V2 (The output of L2) must be TRUE for the output of L3 to be TRUE.
- If the output of L1 and the output of L2 are both TRUE the output of L3 will be FALSE.
- If the output of L1 and the output of L2 are both FALSE the output of L3 will be FALSE.

Logical Switch Tutorial 2 – p6
Logic of XOR for the enthusiast

#	Function	V1	V2	
L1	a=x	SA	100	----
L2	a=x	SB	100	----
L3	AND	L1	!L2	----
L4	AND	!L1	L2	----
L5	OR	L3	L4	----
L6	XOR	L1	L2	----



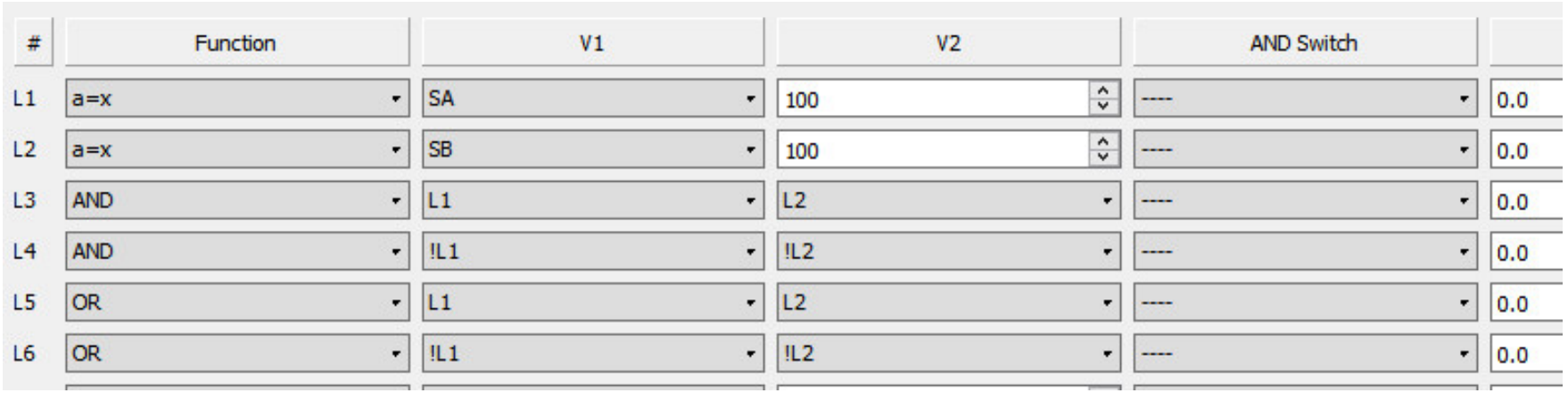
Logical Switch Tutorial 2 – p7
Equivalent AND and OR Functions

This page and the following page is for those who want to investigate the AND and OR Logical Functions some more. You use the appropriate AND or OR function depending on the inputs you put in and the resulting outputs you need.

In Taranis OpenTX you have a choice of using two different versions of any input Value. For example, you can choose either L3 or its opposite !L3 (read as NOT L3) L3 will always output the opposite of !L3.

See the following tables and compare inputs and outputs to see your options.

Logical Switch Tutorial 2 – p8
Equivalent AND and OR Functions



#	Function	V1	V2	AND Switch	
L1	a=x	SA	100	----	0.0
L2	a=x	SB	100	----	0.0
L3	AND	L1	L2	----	0.0
L4	AND	!L1	!L2	----	0.0
L5	OR	L1	L2	----	0.0
L6	OR	!L1	!L2	----	0.0

Table 1 AND

L1	L2	L3	!L3
F	F	F	T
F	T	F	T
T	F	F	T
T	T	T	F

Table 2 AND

!L1	!L2	L4	!L4
T	T	T	F
T	F	F	T
F	T	F	T
F	F	F	T

Table 3 OR

L1	L2	L5	!L5
F	F	F	T
F	T	T	F
T	F	T	F
T	T	T	F

Table 4 OR

!L1	!L2	L6	!L6
T	T	T	F
T	F	T	F
F	T	T	F
F	F	F	T

You are able to build logical functions with either positive (eg. L1) values or negative (eg. !L1) values.

See how an AND result can be equivalent to an OR result depending on whether you use negative or positive inputs and use the negative or positive outputs.

Logical Switch Tutorial 3 - p1
Edge Function

The Edge Function is quite different from the other Logical Switch functions and it is very powerful. There are six different versions, each giving a different timing to the output in response to the input.

The Edge Function responds to a change in state of the input signal, either from OFF to ON (-100 to 100) or from ON to OFF (100 to -100).

If activated it will create a short output pulse if no duration is defined and a longer pulse if a duration is defined.

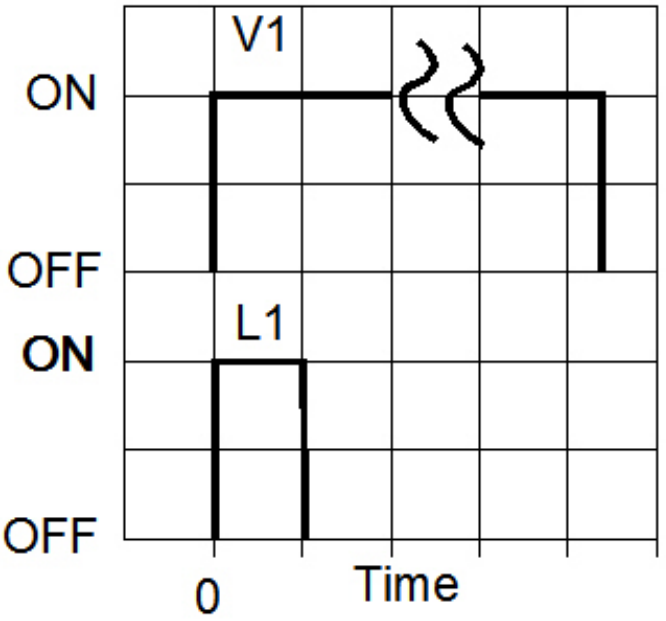
The output can also be delayed if that is defined.

The following Edge Function pages will describe the six variations and some uses of this Logical Switch.

Logical Switch Tutorial 3 - p2
Edge Function – Trigger with Activating signal Activation

#	Function	V1	V2	AND Switch	Duration
L01	Edge	SH↓	0.0	(instant)	----	0.0

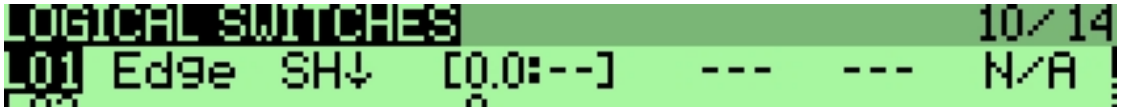
LOGICAL SWITCHES 10/14
L01 Edge SH↓ [0.0:<<] --- --- N/A

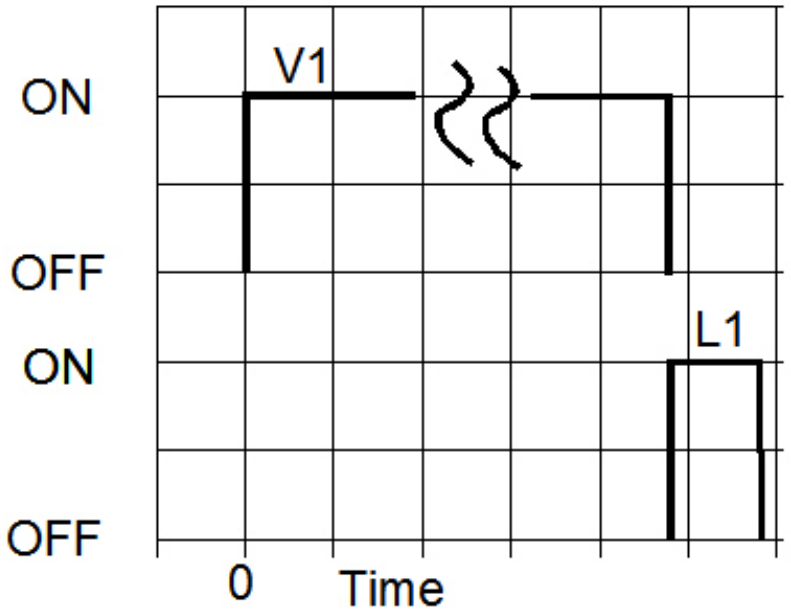


L1 Pulse triggered as soon as the Switch SH is active
Example shows switch V1 with 0.0 sec selected and time is Instant.
The output is one program cycle long unless a longer duration is defined

Logical Switch Tutorial 3 - p3
Edge Function – Trigger with Activating signal Deactivation

#	Function	V1	V2	AND Switch	Duration
L01	Edge	SH↓	0.0	0.0 (infinite)	----	0.0

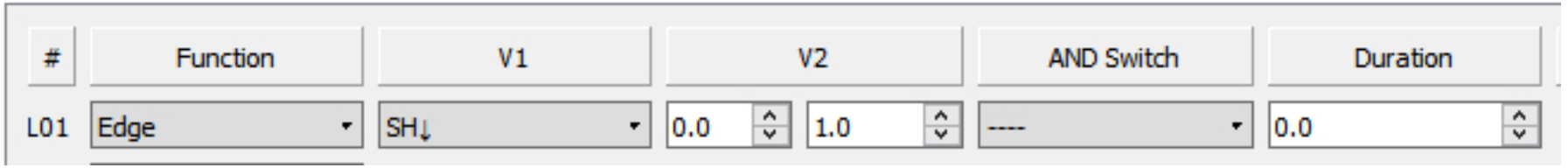


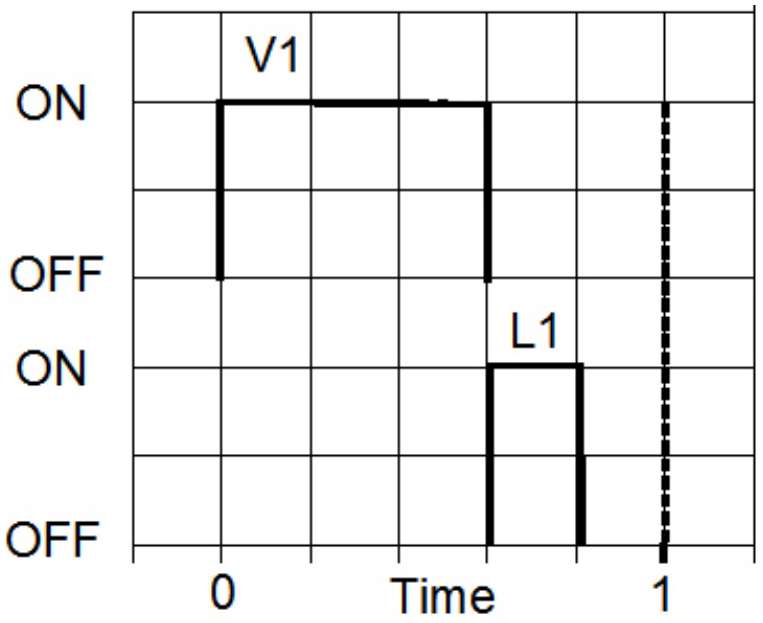


L1 Pulse triggered as soon as the switch SH is Released
Example shows switch V1 with 0.0 sec selected and time is Release.

The output L1 is only one program cycle long unless a longer duration is defined.

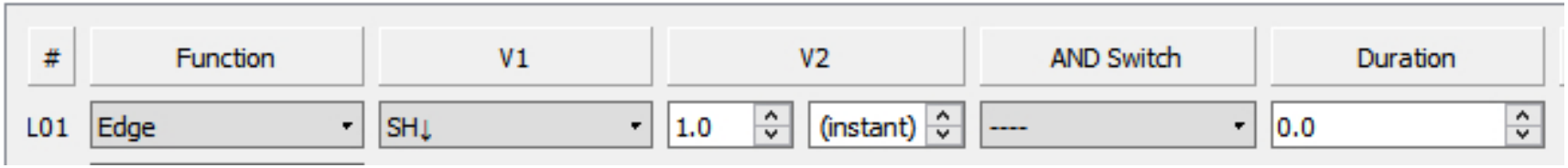
Logical Switch Tutorial 3 - p4
Edge Function – Trigger before the Specified Boundary

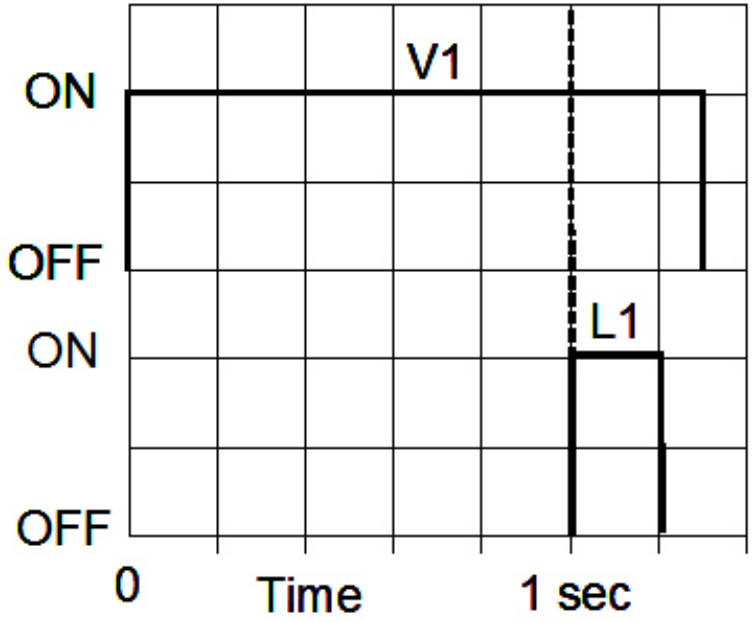




Pulse triggered as soon as Switch SH is Released IF it is released Before 1.0 sec. (The dotted line indicates the time boundary.)
Example shows switch V1 with 0.0 sec selected and Release time is 1.0 sec.

Logical Switch Tutorial 3 - p5
Edge Function – Trigger At the Specified Boundary



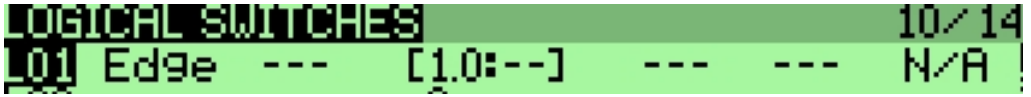


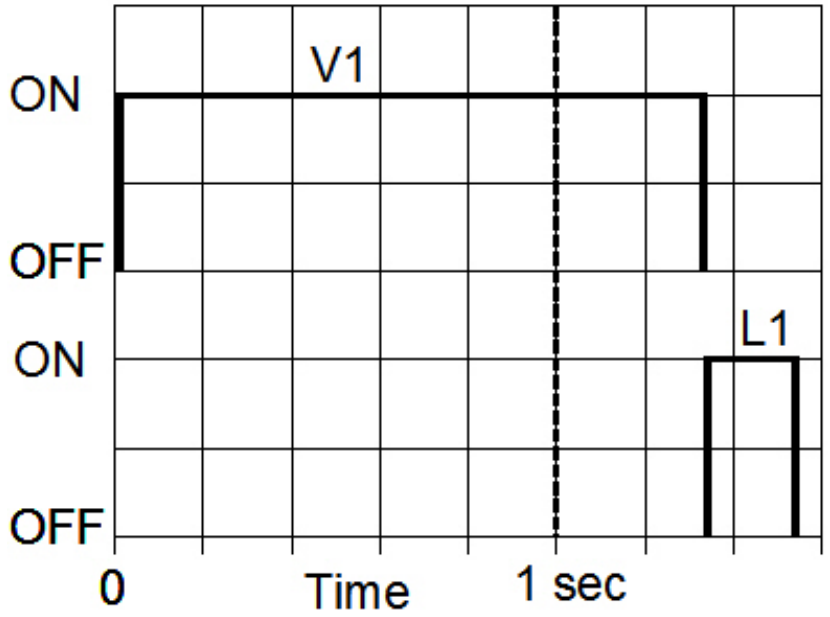
L1 Pulse triggered as soon as Switch SH is Held until the selected 1.0 sec time is passed. (The dotted line indicates the timeout goal.) SH can be held past the timeout goal but L1 will be triggered at the timeout goal

Example shows switch V1 with 1.0 sec selected as the timeout goal and instant as the response when the timeout is reached.

Logical Switch Tutorial 3 - p6
Edge Function – Trigger past the Specified Boundary

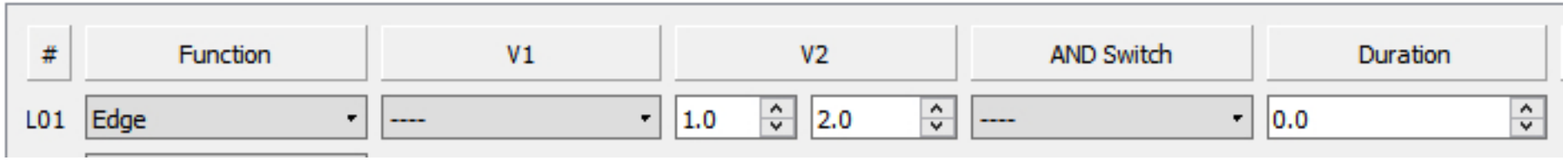
#	Function	V1	V2	AND Switch	Duration
L01	Edge	SH↓	1.0	1.0 (infinite)	----	0.0

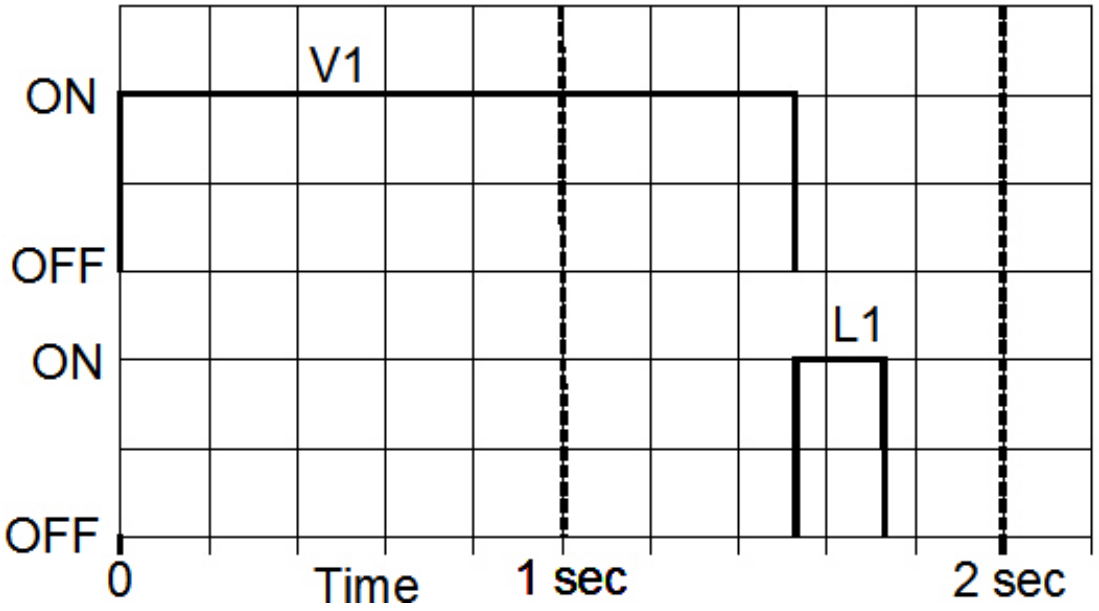




L1 Pulse triggered when Switch SH is Released IF the selected 1.0 sec time is passed. (The dotted line indicates the timeout goal.)
Example shows switch V1 with 1.0 sec selected as the timeout goal and Switch Release as the time when the Logical Switch is activated IF the selected switch is held past the timeout goal.

Logical Switch Tutorial 3 - p7
Edge Function – Trigger between two Specified Boundaries

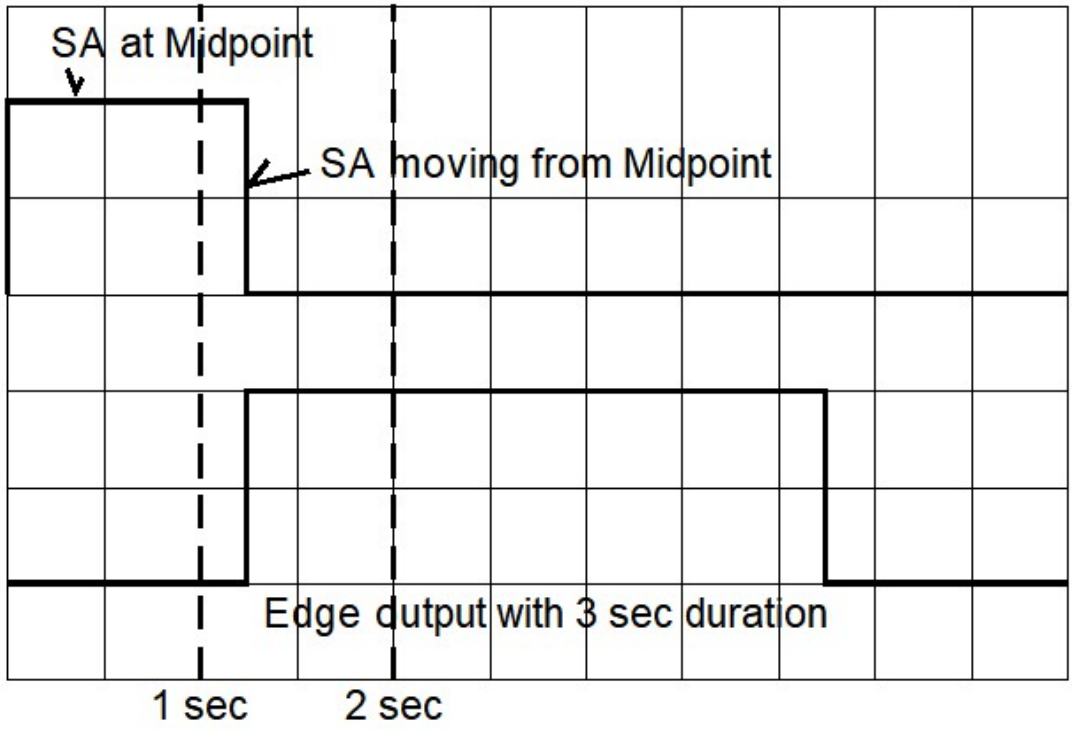




L1 Pulse triggered when the Switch SH is Released After the selected 1.0 sec boundary time is passed AND Before the Second boundary time is reached. (The dotted lines indicates the timeout boundaries.) Example shows switch V1 with 1.0 sec selected as the first boundary, 2.0 sec as the second boundary and Switch Release as the time when the Logical Switch is released some time between the boundaries. If the Switch is released before the first boundary or after the second boundary the Logical Switch will not be triggered.

Logical Switch Tutorial 3 - p8
Edge Function – with Duration

L18	a=x	SA	0		----	0.0	0
L19	Edge	L18	1.0	2.0	----	3.0	
L20	Edge	L18	1.0	(instant)	----	0.0	



All the Edge functions are described as though the duration value is 0.0 seconds. The V1 and V2 values define the time when the Edge output is triggered. The Length of time the output is ON is controlled by the Duration value.

I have used L20 to indicate to me the 1 second boundary of the function when I was trying the function.

Logical Switch Tutorial 3 - p9
Single Channel Compound Escapement Example

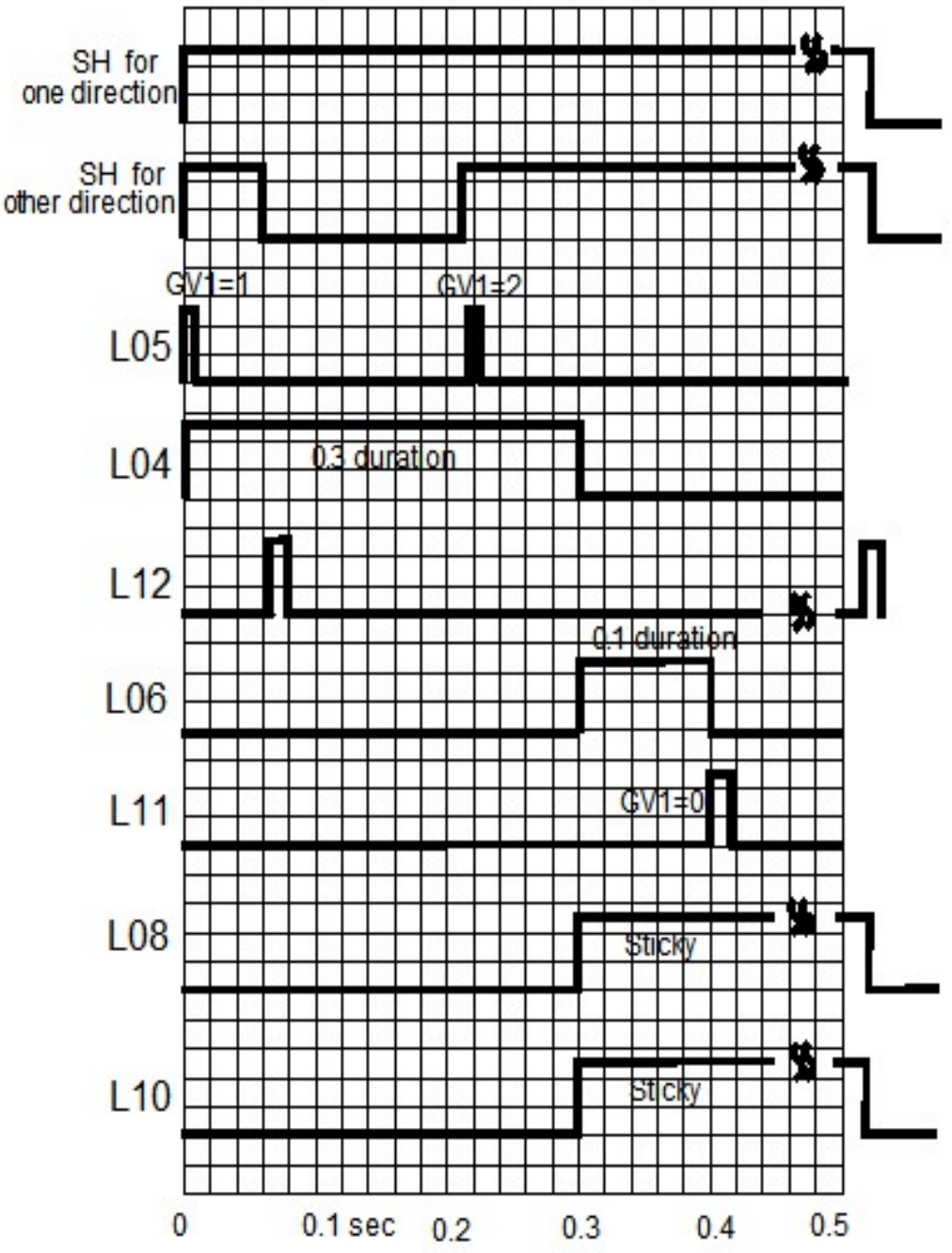
These Logical Functions implement a Single Channel compound Escapement Emulation.
A pull and hold of Momentary Switch SH will move the rudder one way.
Two quick pulls in succession will move the rudder in the opposite direction.

#	Function	V1	V2		AND Switch	Duration
L01	a=x	GV1	1		----	0.0
L02	a=x	GV1	2		----	0.0
L03	a>x	SH	95		----	0.0
L04	Edge	SH↓	0.0	(instant)	----	0.2
L05	Edge	SH↓	0.0	(instant)	----	0.0
L06	Edge	L04	0.0	0.0 (infinite)	----	0.1
L07	AND	SH↓	L06		L01	0.0
L08	Sticky	L07	L12		----	0.0
L09	AND	SH↓	L06		L02	0.0
L10	Sticky	L09	L12		----	0.0
L11	Edge	L06	0.0	0.0 (infinite)	----	0.0
L12	Edge	L03	0.0	0.0 (infinite)	----	0.0

Logical Switch Tutorial 3 – p10
Edge Function
Compound Escapement
Emulation Example
Timing Diagrams

Notice – Logical Switch outputs can be inputs for other Logical Switches so you can make Complex timing sequences.

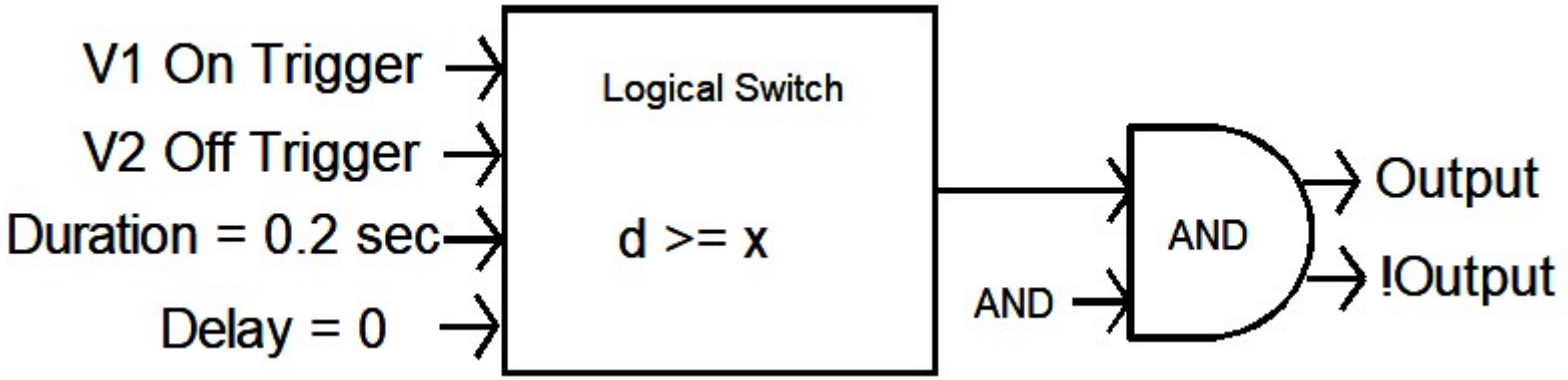
Momentary On Switch SH is pulled once or twice to start the Rudder movement
The first pull of SH does two things.
It activates L5 to increment GV1 and starts L4 creating a pulse 0.3 sec. long.
The end of L4 sets either Sticky L08 or L10 to move the rudder, depending on GV1 having a value of 1 or 2.
End of L06 triggers L11 to reset GV1 to 0.
Release of SH pulses L12 to reset the Sticky of L08 or L10 to return the rudder to center position.



Logical Switch Tutorial 4 – p1
d >= x

The software checks input VI every 30 ms to see if it has changed by at least V2. If V2 is positive, it looks for a positive change. If negative it looks for a negative change. If such a change is seen, it signals a TRUE Logical Switch output

#	Function	V1	V2	AND Switch	Duration	Delay
L1	d>=x	S1	-30	----	0.2	0.0



A negative change of -30 in Rotary Potentiometer S1 setting sets L1 true

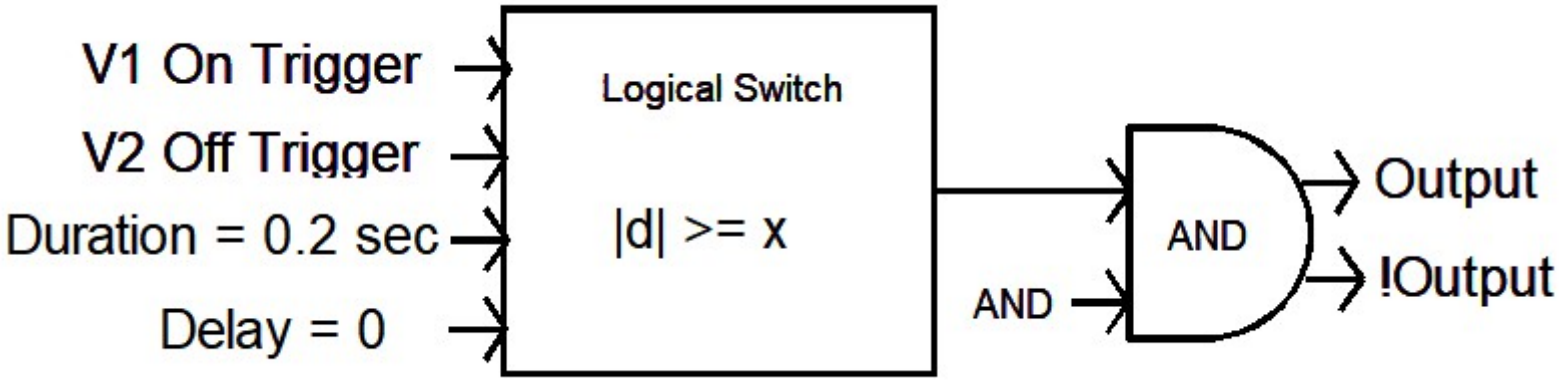
If Duration = 0.0, L1 is true for one cycle of the program

If Duration value is entered (eg 0.2 sec) then L1 will be true for Duration

Logical Switch Tutorial 4 – p2
|d| >= x

The software checks input VI every 30 ms to see if it has changed by at least the magnitude of V2, either positive or negative. If so it signals a TRUE Logical Switch output

#	Function	V1	V2	AND Switch	D
L1	|d|>=x	S1	30	----	0.2



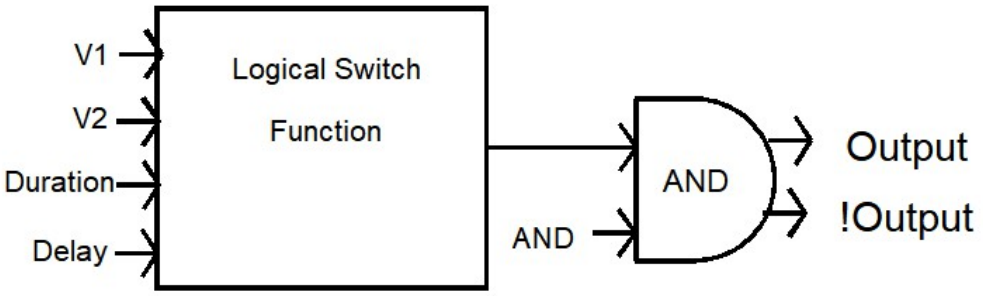
A change of 30 in either direction of Rotary Potentiometer S1 sets L1 TRUE

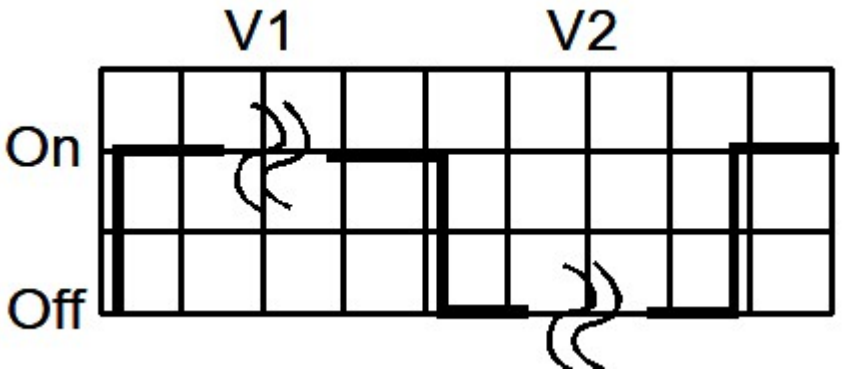
If Duration = 0.0, L1 is true for one cycle of the program

If Duration value is entered (eg 0.2 sec) then L1 will be true for Duration

Logical Switch Tutorial 5 – p1
Timer

#	Function	V1	V2	AND Switch	Duration	Delay
L01	Timer	3.5	0.5	!SA-	1.0	0.5





V1 sets the ON time of the timer – 3.5 sec
V2 sets the OFF time of the timer – 0.5 sec
!SA– is used as a gate to start and stop the timer output
Delay delays the start of the ON time within the total cycle time – 0.5 sec delay
Duration defines the ON time within the total cycle time – 1.0 sec ON time

Logical Switch Tutorial 5 – p2
Possible Timer Use

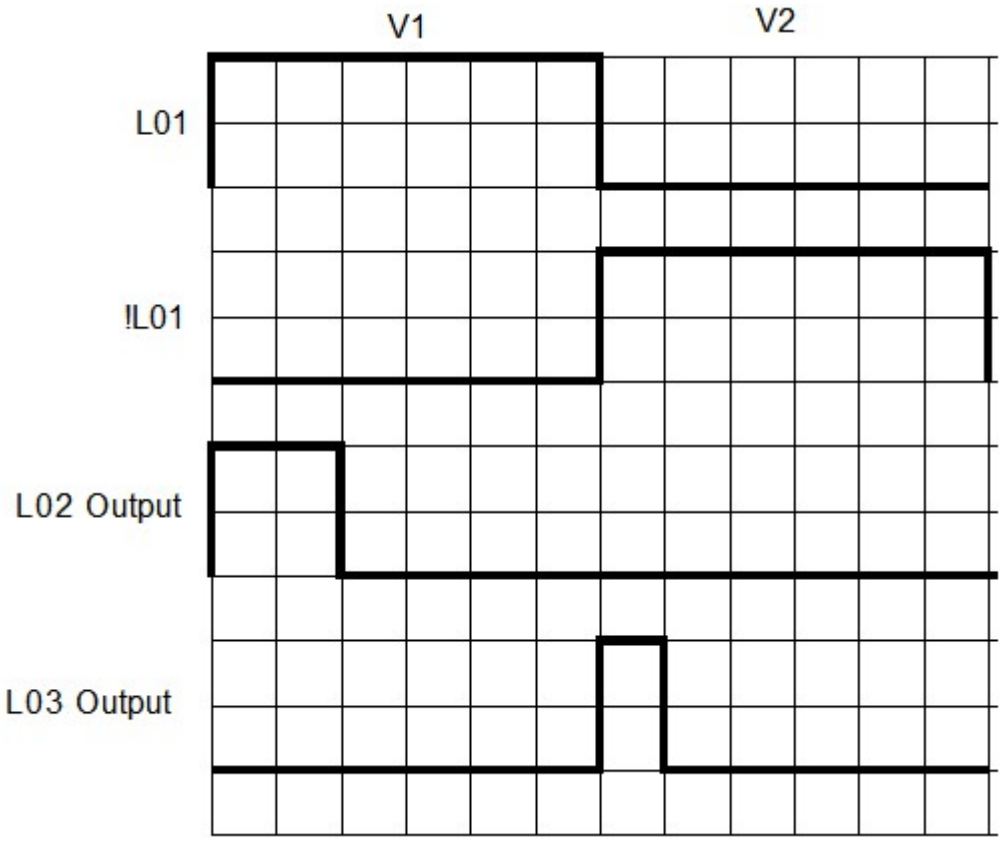
#	Function	V1	V2	AND Switch
L01	Timer	3.0	3.0	SA↑
L02	Edge	L01	1.0 (instant)	----
L03	Edge	!L01	0.5 (instant)	----

Create two repeating pulses of different lengths

V1 sets the ON time of the timer (3.0 sec)
V2 sets the OFF time of the timer
SA starts and stops the timer output

The leading edge of L01 triggers the L02 Edge function to create a pulse
The leading edge of !L01 triggers the L03 Edge function to create a different length pulse.

Notice, the Edge function can be triggered by either the L01 or the !L01 output.



Logical Switch Tutorial 5 – p3
Possible Timer Use – using Duration and Delay Values

#	Function	V1	V2	AND Switch	Duration	Delay
L01	Timer	3.0	0.1	----	0.2	0.0
L02	Timer	3.0	0.1	----	0.2	0.2
L03	Timer	3.0	0.1	----	0.2	0.4
L04	Timer	3.0	0.1	----	0.2	0.8
L05	Timer	3.0	0.1	----	0.2	1.2
L06	Timer	3.0	0.1	----	0.2	1.6
L07	Timer	3.0	0.1	----	0.2	2.0
L08	Timer	3.0	0.1	----	0.2	2.2
L09	Timer	3.0	0.1	----	0.2	2.4

#	Switch	Action	Parameters	Enable
SF1	L01	Play Sound	Beep 1	No repeat
SF2	L02	Play Sound	Beep 1	No repeat
SF3	L03	Play Sound	Beep 1	No repeat
SF4	L04	Play Sound	Beep 3	No repeat
SF5	L05	Play Sound	Beep 3	No repeat
SF6	L06	Play Sound	Beep 3	No repeat
SF7	L07	Play Sound	Beep 1	No repeat
SF8	L08	Play Sound	Beep 1	No repeat
SF9	L09	Play Sound	Beep 1	No repeat

Send repeating “SOS” in Morse Code

Logical Switch Tutorial 5 – p4
Long Period Timer – Time Lapse Photography?

#	Function	V1	V2	AND Switch	Duration	Delay
L01	Timer	0.9	0.1	!SA-	0.0	0.0
L02	a=x	GV9	10	----	0.0	0.0
L03	a=x	GV9	0	----	0.0	0.0
L04	Sticky	L03	L03	----	0.0	0.0

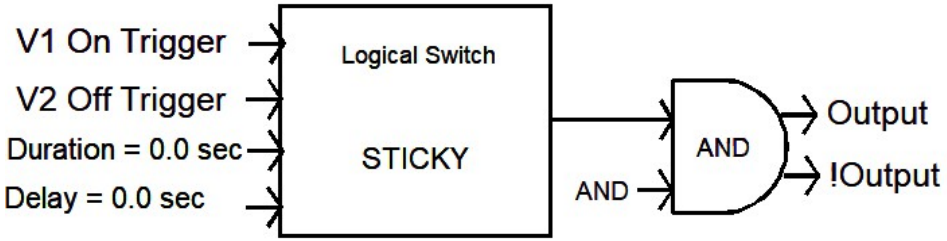
#	Switch	Action	Parameters		Enable
SF1	L01	Adjust GV9	Increment	1	ON
SF2	L02	Adjust GV9	Value	0	ON
SF3	One	Adjust GV9	Value	0	ON
SF4	SB↑	Adjust GV9	Value	0	ON

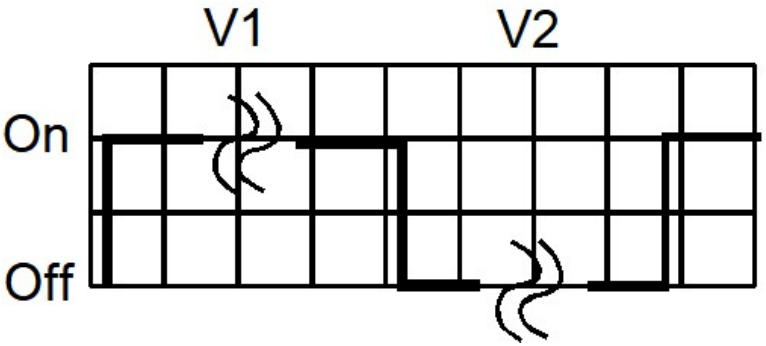
- The Long Period Timer uses the Timer (Logical Switch L01) to increment a GV (SF1).
- L02 goes TRUE when the GV counter reaches the desired Max count.
- L02 sets the GV back to the start count value and Toggles the L04 Sticky.
- The L04 Sticky change of state can be used to Trigger some desired function.

- The Maximum value which can be entered into the Timer Logical Switch V1 and V2 is 175 for a maximum Timer period of 350 sec or almost 6 minutes.
- The Maximum value of a GV is 1024 so the maximum cycle time of L04 can be about 12000 minutes or 8+ days.

Logical Switch Tutorial 6 – p1
Sticky

#	Function	V1	V2	AND Switch	Duration	Delay
L01	Sticky	SA-	!SA-	----	0.0	0.0

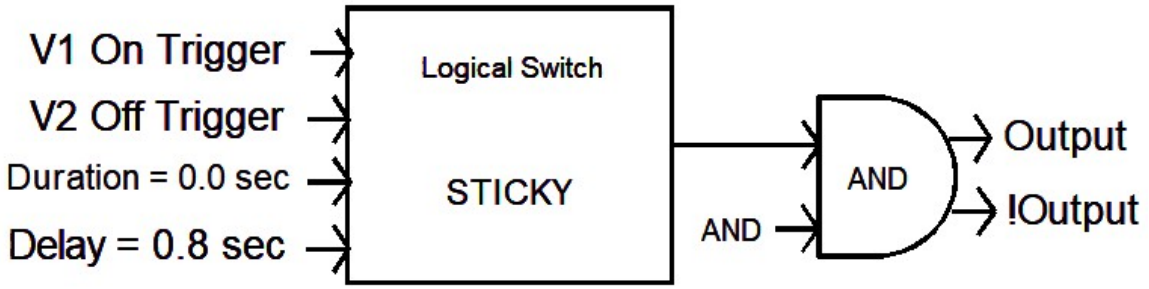


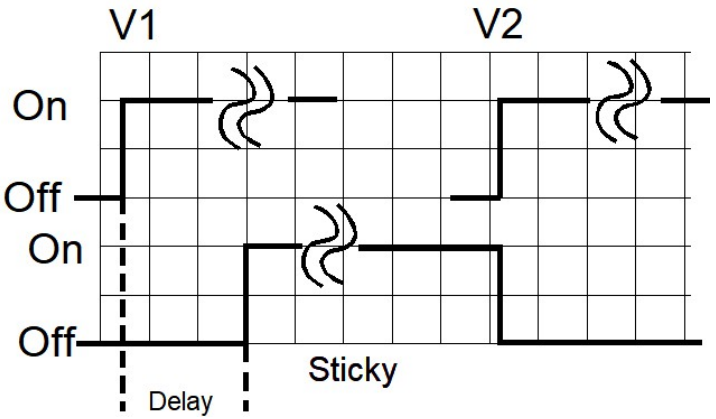


The Sticky Function is like a toggle switch but two commands can control it
It is set ON with leading edge of V1 going TRUE,
It is canceled OFF with leading edge of V2 going TRUE
It ignores any changes to V1 when the sticky is TRUE
It ignores any changes to V2 when the Sticky is FALSE
V1 may or may not be the same source signal as V2

Logical Switch Tutorial 6 – p2
Sticky with delay

#	Function	V1	V2	AND Switch	Duration	Delay
L01	a>x	I3:Thr	-60	----	0.0	0.0
L02	Sticky	L01	SA↑	----	0.0	0.8





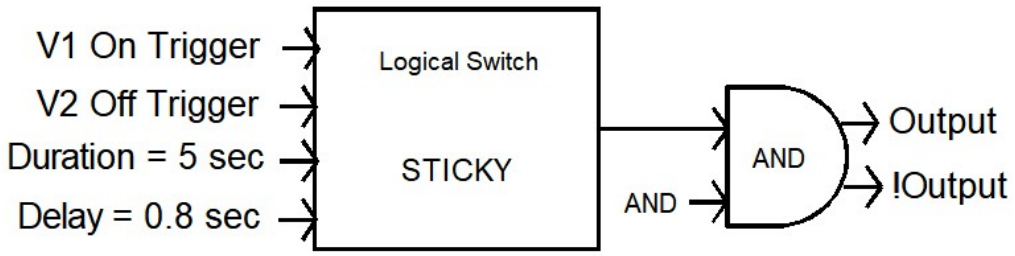
L01 commands the Sticky to set but the Delay delays the set for the Delay time

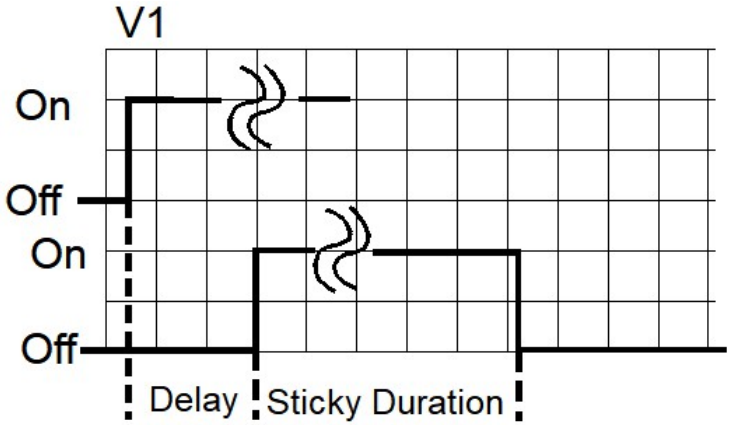
A second command will reset the Sticky.

eg. You want to raise the flaps for a “go-around” but you want to wait until the motor power has been definitely applied.

Logical Switch Tutorial 6 – p3
Sticky with Delay and Duration

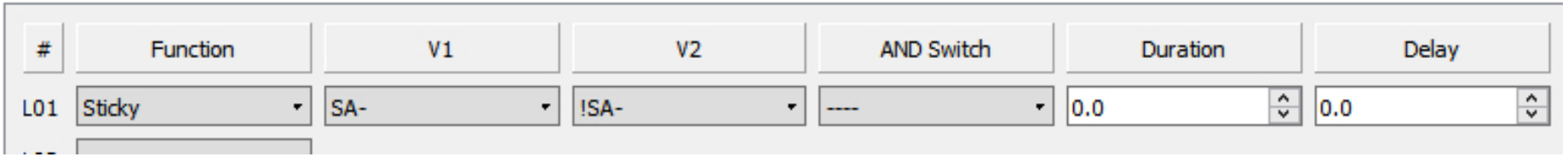
#	Function	V1	V2	AND Switch	Duration	Delay
L01	Sticky	SH↓	----	----	5.0	0.8
L02						

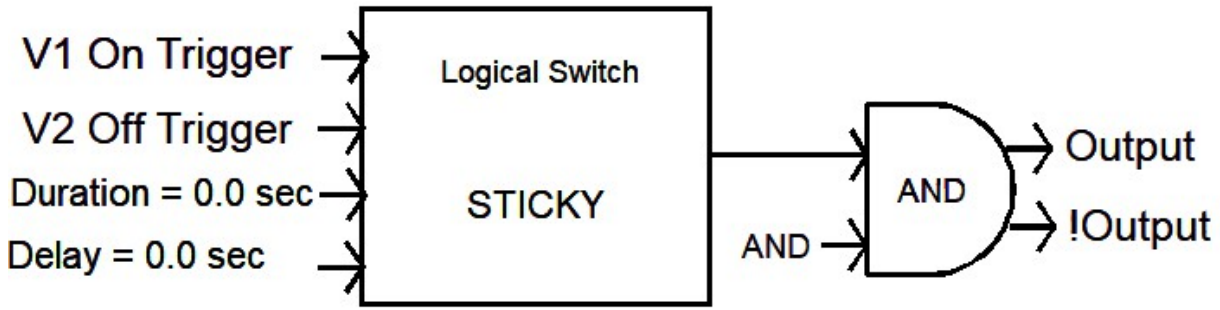




The input signal V1 commands the Sticky to set but the Delay holds the set for the Delay time, then sets the Sticky ON. It stays ON for the time of the Defined Duration.

Logical Switch Tutorial 6 – p4
Warning concerning Sticky Use with AND

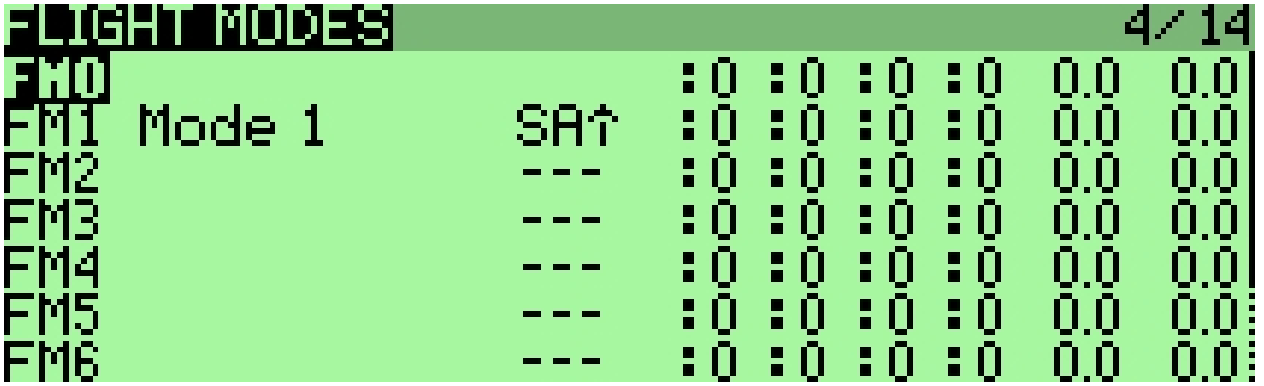




The state of the STICKY function can be hidden by the AND gate.
If you have an input on the AND gate and set that input OFF, the output will not represent the state of the STICKY. This could be dangerous if you are using the STICKY to lock a condition for some reason.

Logical Switch Tutorial 7
Logical Switches in Use

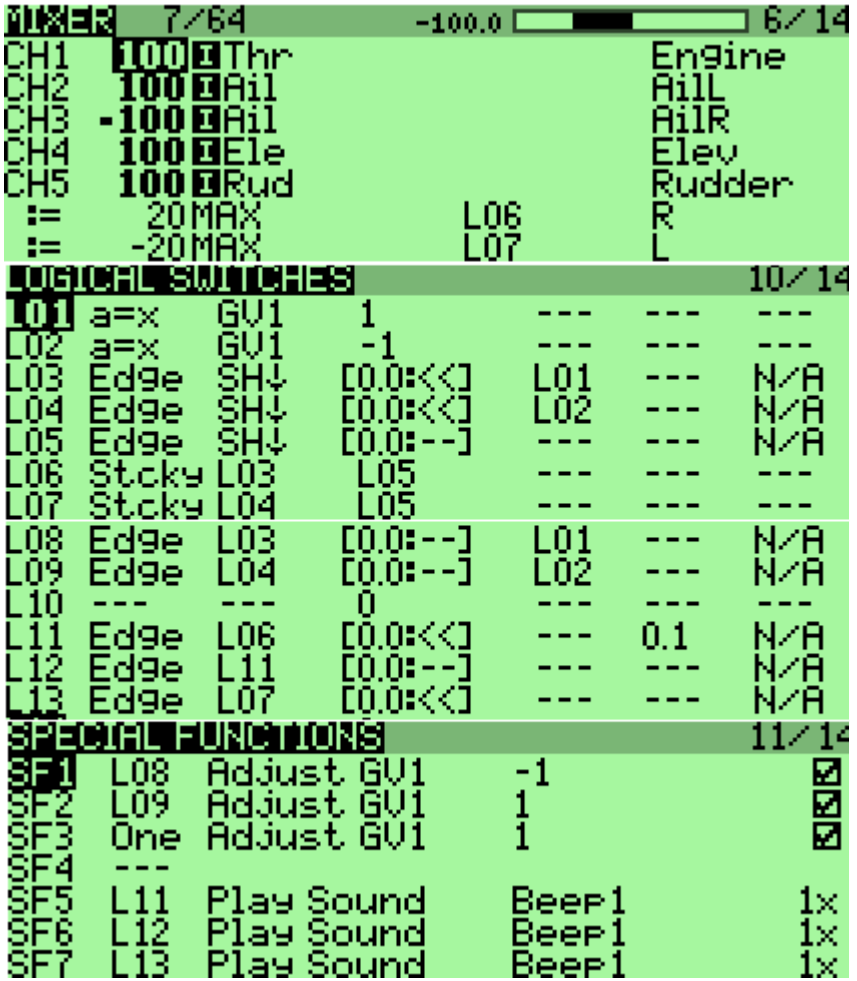
- Both physical switches (SA etc) and Logical Switches (L1 etc) are used to create “Flags” for the computer to notice regularly. When it sees a Flag (a switch position, or state) it checks to see what it should do when that Flag is encountered and does the specified action.
- For example, a simple task would be to select a Flight Mode. Simply select physical switch SA↑ to activate FM1 and when you move switch SA to SA↑ the system will switch to Flight Mode 1.



- In the same way we can use the Logical Switches to signal actions we want to do. The following examples will show Logical Switches creating variables to manipulate and signal pulses to trigger other signal pulses to command various actions.

Logical Switch Tutorial 7 – Example 1, p1
Simple Escapement Emulation

- A simple escapement was used to control the rudder of the earliest RC airplanes. One press and hold of the control button caused the rudder to swing to one side, a release of the button would bring the rudder back to neutral. The next press and hold caused the rudder to swing to the other side, release would return it. The next press and hold would repeat the process. We can emulate this action using the momentary switch SH and Logical Switches.



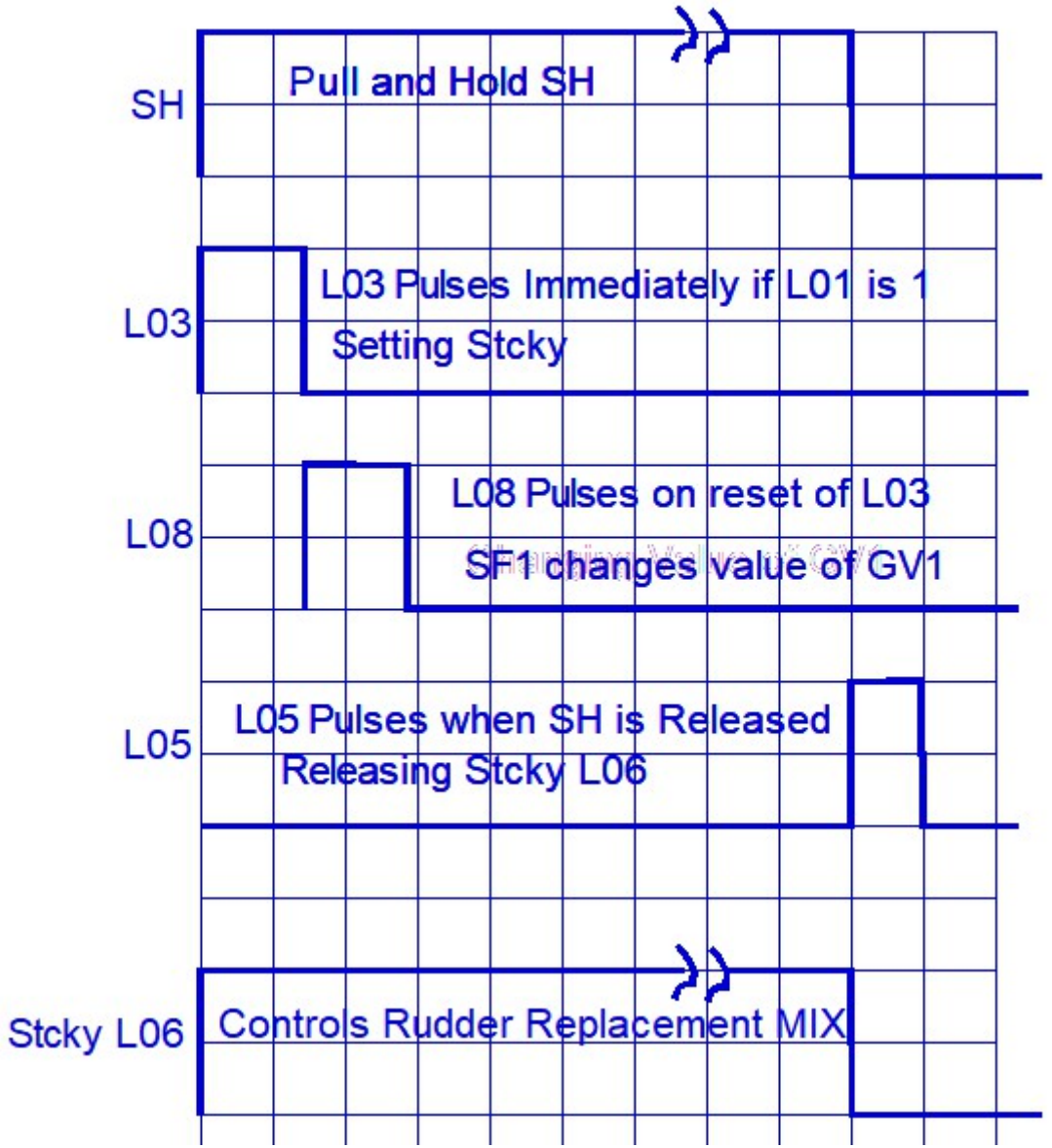
Logical Switch Tutorial 7 – Example 1, p2
Simple Escapement Emulation (continued)

- L01 and L02 create two flags to watch the value of GV1 as the program proceeds.
- L03 and L04 are Edge Functions watching for Switch SH to be pulled ON.
- IF GV1 has the value 1 (L01 is active) when SH is pulled ON, the L03 Edge Function will go active with a short pulse . This will SET the L06 Stcky Function which will select the Ch5 replacement 20 MAX mix to move the rudder.
- IF GV1 has the value -1 (L02 is active) when SH is pulled ON, the L04 Edge Function will go active . This will SET the L07 Stcky Function which will then select the Ch5 replacement -20 MAX mix to move the rudder the other way.
- L05 is an Edge Function watching for the Switch SH to be released. When SH is released L05 will go active for a short time, signaling either L06 or L07 Stcky to deactivate, which tells the Rudder Mix to command the rudder to return to neutral.

Logical Switch Tutorial 7 – Example 1, p3
Simple Escapement Emulation (continued)

- L01 and L02 create two flags to watch the value of GV1 as the program proceeds.
- L03 and L04 are Edge Functions watching for Switch SH to be pulled ON.
- IF GV1 has the value 1 (L01 is active) when SH is pulled ON, the L03 Edge Function will go active with a short pulse . This will SET the L06 Stcky Function which will select the Ch5 replacement 20 MAX mix to move the rudder.
- IF GV1 has the value -1 (L02 is active) when SH is pulled ON, the L04 Edge Function will go active . This will SET the L07 Stcky Function which will then select the Ch5 replacement -20 MAX mix to move the rudder the other way.
- L05 is an Edge Function watching for the Switch SH to be released. When SH is released L05 will go active for a short time, signaling either L06 or L07 Stcky to deactivate, which tells the Rudder Mix to command the rudder to return to neutral.

Logical Switch Tutorial 7 – Example 1, p4
Simple Escapement Emulation (continued)
Timing diagram

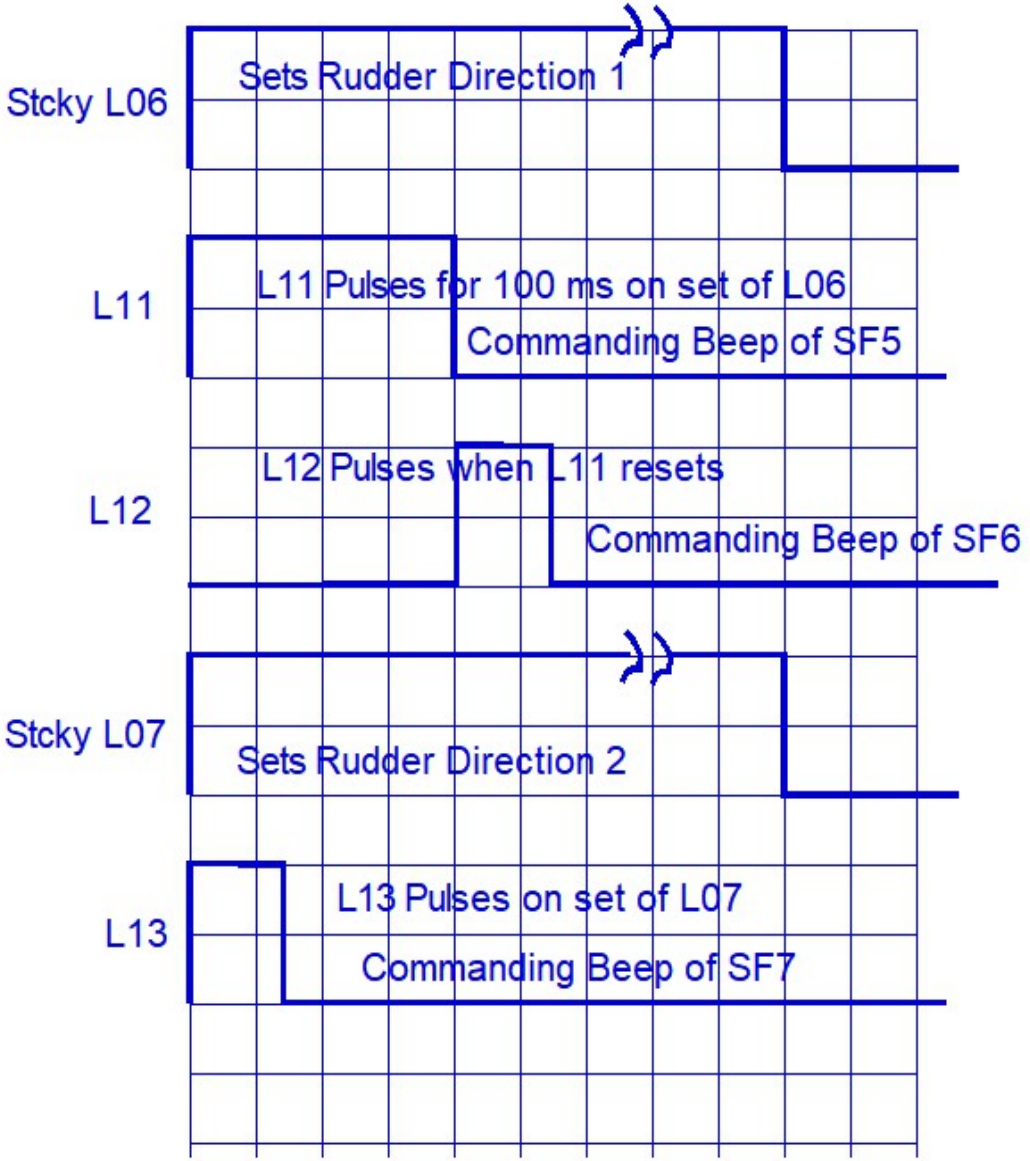


Logical Switch Tutorial 7 – Example 1, p5
Simple Escapement Emulation (continued)

- But the direction of the Rudder movement is controlled by the value of GV1. How is the value of GV1 changed?
- SF1, SF2, and SF3 change the value of GV1 on command. L08 going active changes the value of GV1 to -1. L09 going active changes the value of GV1 to 1. The Special Function command “One” sets GV1 to 1 when the model is first activated so the function has a place to start.
- L08 is an Edge Function watching for the falling edge of the L03 pulse. When the falling edge of L03 is detected L08 creates a short pulse which changes the value of GV1 to -1. This change is made after the Stcky function is set because the old value just used is not needed any more but before the Stcky is reset making it ready for the next rudder selection.
- L09 is an Edge Function watching for the falling edge of the L04 pulse. When the falling edge of L04 is detected L09 creates a pulse which changes the value of GV1 to 1.

Logical Switch Tutorial 7 – Example 1, p6
Simple Escapement Emulation (continued)
Double Beeps

- We want an audio indication of the Rudder movement as well. It would be nice to hear a single beep for one direction and a double beep for the opposite direction.
- Detect L06 Stcky SET and beep twice.
- Detect L07 Stcky SET and beep once.
- L11 watches for L06 to set, triggers SF5 to beep and makes a 100 ms long pulse.
- L12 watches for the end of the L11 pulse and triggers SF6 to beep. You hear two beeps.
- L13 watches for L07 to SET, triggers SF7 to beep. You hear a single beep.



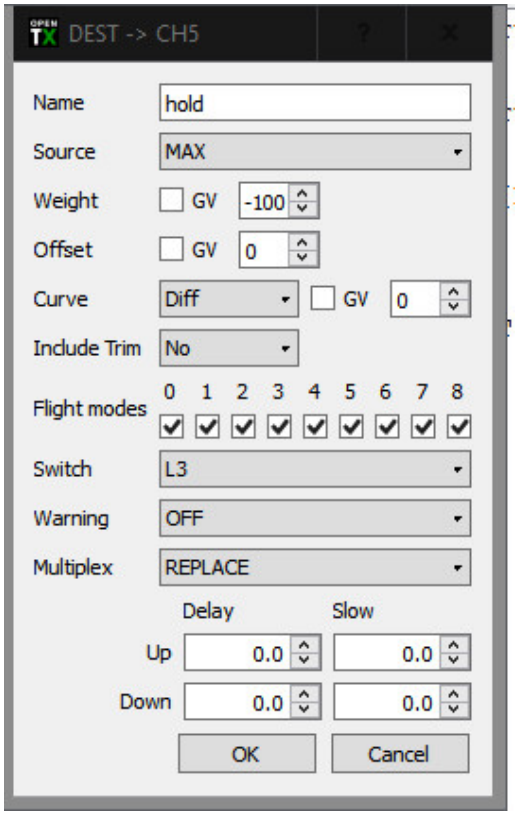
Logical Switch Tutorial 7 – Example 2, p1
Throttle Lock-out Example

- A Logical Switch can also be used to select a change in system configuration. Consider the Throttle Lock-Out below.

Mix

```
CH05            [I3]Thr Weight(+100%)
              := MAX Weight(-100%) Switch(L3) NoTrim [hold]
```

Mix Definition Table



Logical Switch Definition Table

#	Function	V1	V2		
L1	a<x	[I3]Thr	-98		----
L2	Edge	SH↓	0.8	(instant)	L1
L3	Sticky	L2	L2		----
L4	Edge	SH↓	0.0	0.7	L3

Special Function Definition Table

#	Switch	Action	Parameters	
SF1	L2	Play Sound	Beep 3	No repeat
SF2	L4	Play Track	thrhold ▶	No repeat

Logical Switch Tutorial 7 – Example 2, p2
Throttle Lock-out Example (continued)

- A Replacement Mix is defined for the Throttle. This is activated when the L03 Logical Switch is seen TRUE.
- Logical Switch L01 is TRUE when Throttle has a value less than -95 and stays TRUE until the Throttle value becomes greater than -96.
- Logical Switch L02 is set by an EDGE function. When the momentary switch SH is pulled (SH↓) it waits for 0.8 seconds before it sets L02 True for one cycle of the processor time. (approx. 30 msec). If SH is released before 0.8 sec. L02 is not set. L2 is only set TRUE if L1 is TRUE
- Logical Switch L03 is a Sticky, set to TRUE when it sees L2 set TRUE. L03 then waits for the next time L02 is set TRUE to set L03 back to FALSE.
- Logical Switch L04 is an edge function that is set to TRUE for once cycle if SH↓ is released before 0.7 sec if L03 is TRUE. This causes a Special Function Voice warning when TRUE.
- Logical Switch L02 causes a Special Function audible beep if L2 becomes TRUE.
- Logical Switch L03 replaces the continuous Throttle Mix with a MAX = -100 Mix
- Logical Switch L04 causes a Special Function Voice warning if L4 becomes TRUE.

- The Logical Switches are used in quite different ways.

  L01 just watches for a particular state of the Throttle and signals when it is seen.

  L02 watches the state of SH and signals at a specified time.

  L03 locks a particular state of its signal until it is deliberately told to unlock it. It commands a change of state of the Throttle

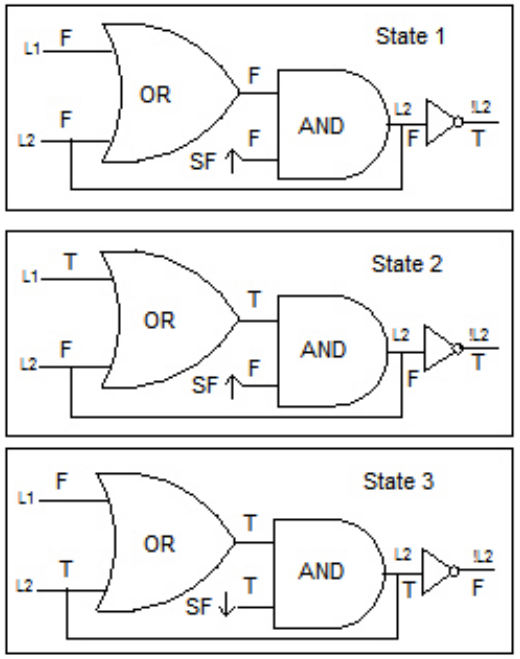
  L04 signals to activate a Special Function voice warning.

Logical Switch Tutorial 7 – Example 3, p1
Another Throttle Lock-out Example

Here is another very interesting Throttle Lock-Out. It uses the L2 output as feedback to an L2 input. Note: !L2 controls the Mix replacement Function

Mix

```
CH03              [I3]Thr Weight(+100%)
                := MAX Weight(-100%) Switch(!L2) [cutoff]
```



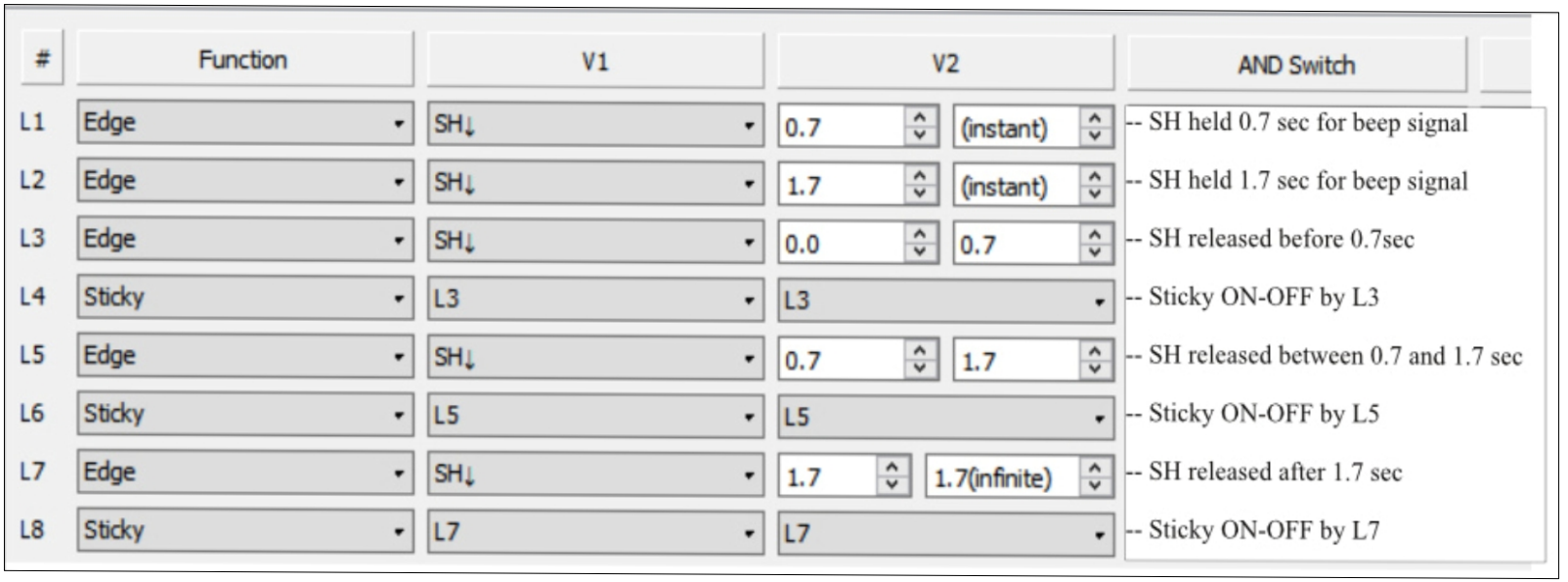
Logical Switch Definition Table

#	Function	V1	V2	AND Switch
L1	a<x	[I3]Thr	-98	----
L2	OR	L2	L1	SF↓

- State 1: Throttle > -98 so L1 is FALSE, SF↓ is FALSE so L2 is FALSE and !L2 is TRUE holding the Throttle at -100
- State 2: Throttle < 98 making L1 TRUE but SF↓ is FALSE so !L2 is TRUE holding Throttle still at -100
- State 3: SF↓ is TRUE so OR TRUE output can pass through the AND making !L2 FALSE. The cutoff Mix is disabled allowing the Throttle changes to be active
- Notice the use of the !L2 Logic output. See Page 15

Logical Switch Tutorial 7 – Example 4, p1
Multiple Commands on one Momentary ON Switch

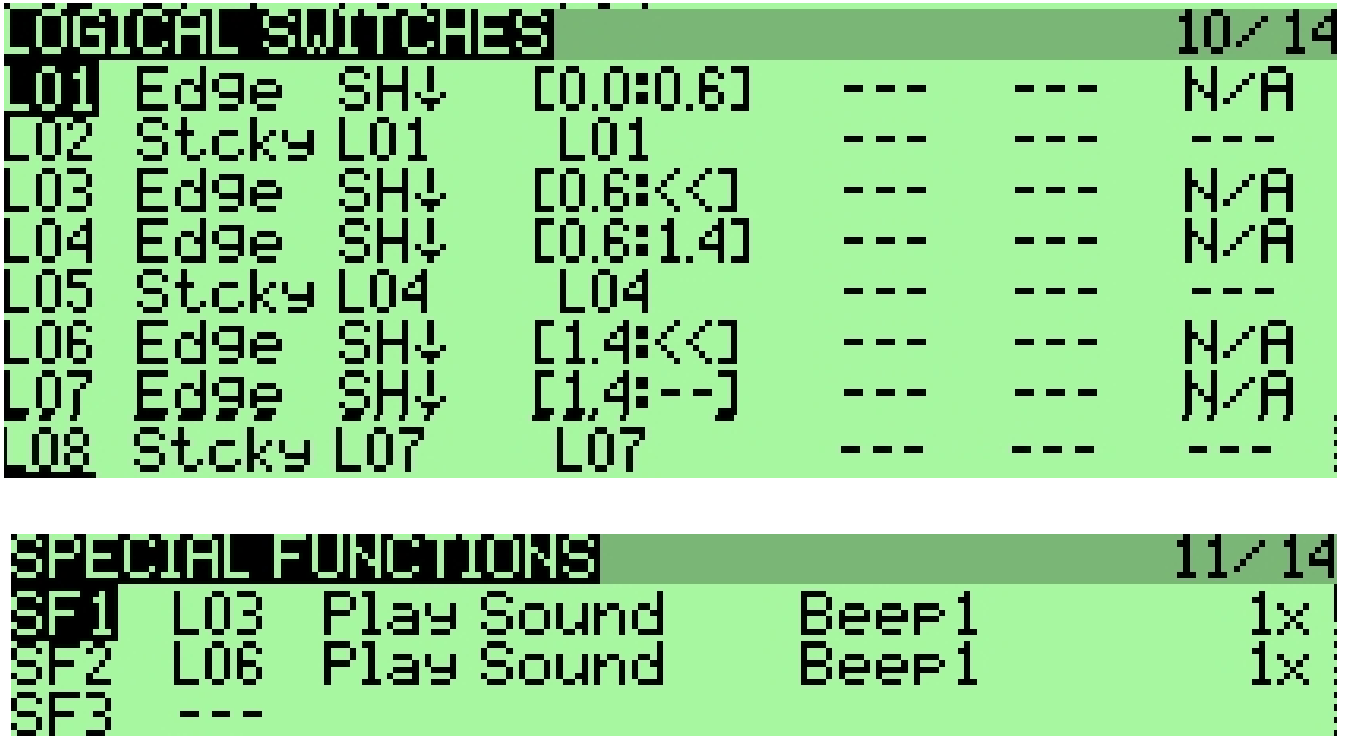
This function enables the Single Throw Momentary Switch SH
to independently command three Logical Switches ON and OFF



#	Function	V1	V2	AND Switch	
L1	Edge	SH↓	0.7	(instant)	-- SH held 0.7 sec for beep signal
L2	Edge	SH↓	1.7	(instant)	-- SH held 1.7 sec for beep signal
L3	Edge	SH↓	0.0	0.7	-- SH released before 0.7sec
L4	Sticky	L3	L3		-- Sticky ON-OFF by L3
L5	Edge	SH↓	0.7	1.7	-- SH released between 0.7 and 1.7 sec
L6	Sticky	L5	L5		-- Sticky ON-OFF by L5
L7	Edge	SH↓	1.7	1.7(infinite)	-- SH released after 1.7 sec
L8	Sticky	L7	L7		-- Sticky ON-OFF by L7

L1 waits for 0.7 sec then is active for one OTX cycle, triggering a beep Special Function
L2 waits for 1.7 sec then is active for one OTX cycle, triggering a beep Special Function
L3 is active before 0.7 sec for one OTX cycle if SH is released before 0.7 sec beep
L4 Sticky is set or reset by L3 pulse
L5 is active after 0.7 sec and before 1.7 sec for one OTX cycle if SH is released after the 0.7 sec beep and before 1.7 sec. beep
L6 Sticky is set or reset by L5 pulse
L7 is active after 1.7 sec for one OTX cycle if SH is released after the 1.7 sec beep
L7 Sticky is set or reset by L6 pulse

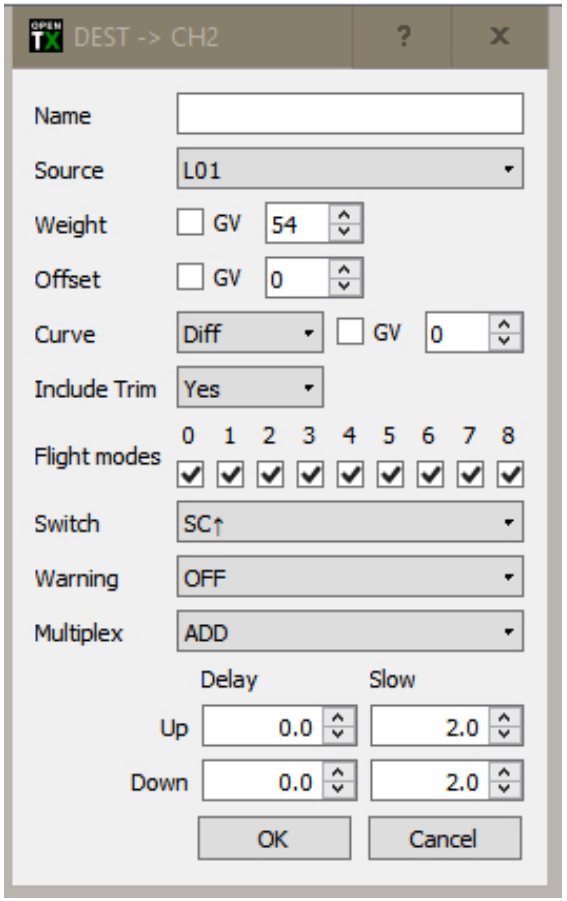
Logical Switch Tutorial 7 – Example 4, p2
Multiple Commands on one Momentary ON Switch (continued)

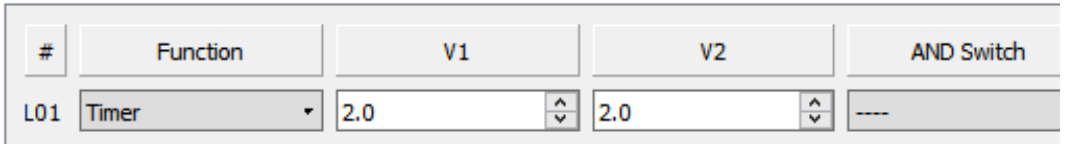


- Three independent Stcky Logical Switches can be set and reset independently.
- One quick pull and release of SH sets or resets L02
- Pull and wait for first beep then release SH to set or reset L05
- Pull and wait for the second beep then release SH to set or reset L05

- More than three commands can be cascaded in the same way.

Logical Switch Tutorial 7 – Example 5, p1
Timer Logical Switch and Range





Sometimes it is difficult to range test your model if you are alone at the field. You must separate the model and TX to do the range test.
If you use the Timer Logical Switch to slowly move a control surface from one extreme to another while you have the TX in Range check mode you can put the TX on the table and carry the model with you away from the TX. Now you will be able to observe the effect of the increasing distance from your TX by watching the moving control surface. In this example you use the switch SC to command the timer to start moving the control surface.