

Logical Switches

I'm your guide, Jenny, and in this article we'll discuss the various options available in the Logical Switches tab of the Model Setup page. If you haven't worked with logic functions before, they can be quite confusing. Hopefully, this effort will help you understand what's going on with logical switches.

Let's start with the basics...

What are Logical switches?

Logical switches are user programmed switches. They aren't real switches that you flip from one position to another, however they can be used as triggers just as any physical switch can be. Rather than being turned on or off by the action of physically adjusting a switch, they are turned on and off by evaluating the conditions of the programming. This may use a variety of inputs such as physical switches, logical switches, sources (channel values, telemetry values, timer values, or Global Variables). They can even use values returned by a LUA model script (LUA1a is the first output from the script selected at LUA1 on the custom scripts page, LUA1b is the second output from the same script, LUA2a is the first output from LUA2 etc) etc.

Logical Switches are set by choosing the **Function**, then refining the options (or parameters): **V1**, **V2**, **AND switch**, **Duration**, and a **Delay** for each switch.



AND Switch, Duration and Delay

The AND switch, duration and delay for the switches work in the same for all switch functions, so we'll have a look at them first.

AND Switch: Select any physical switch, logical switch or Flight mode (ie. SA↓ or L1 or FM1) or a not switch (ie !SA↓ or !L1 or !FM1 meaning not SA↓, Not L1 and Not FM1 respectively) from those available under the “AND switch” options and the value it must be in V2. Only if this is true (on) AND the rest of the switch conditions are true will the switch be on. The Switch function V1 & V2 are evaluated FIRST, then the AND switch applied afterwards. This is important to remember, particularly with the Sticky.

Duration: The length of time the switch will stay ON. If set to 0.0, the switch will remain on until the conditions make the switch off. Any other setting will cause the switch to go off after the number of seconds selected, even if the conditions remain true.

Delay: is the a delay before the switch comes on once the conditions are true

Logical Switch Functions

So what do each of the functions do? The examples here are only designed to demonstrate how the switch works.

NOTE: In the definitions below, the Companion screen is on the left and the Taranis screen on the right.

a=x

This is used to check if the value of **a** (a selectable source) is equal to **x** (a chosen value). When using this switch, the options are:

Function: **a=x**

V1: the source to be used as **a**

V2: the value for **x**

Using Thr as V1 and 0 as V2 means that the switch is true when the throttle output is exactly at it's midpoint (value = 0).

This could be used to trigger, for instance, compensation for spoiler deployment in a glider.



a~x

This is used to check if the value of **a** (a selectable source) is approximately equal to **x** (a chosen value) where "approximately" means equal to or + or - 0.9 from the set value. In other words, the switch will be true if the value of **a** (selected in V1) is within +/- 0.9 of whatever value you set for **x** in V2. If you set x=50, then the switch would be true if whatever was selected as a source in V1 was returning a value of between 49.1 and 50.1. When using this switch, the options are:

Function: **a~x**

V1: the source to be used as **a**

V2: the value for **x**

Using Thr as V1 and 0 as V2, the switch is "true" (ON) if V1 is returned as a value between -0.9 and +0.9, or just a bit plus or minus of it's center position.

Remember, sticks provide values between -100 and +100, so the midpoint is 0. If you had selected $x = 50$, then the switch would be true when you were at approximately three quarter throttle because that's halfway between center (value = 0) and full throttle (value = 100).



a>x

This is used to check if the value of **a** (a selectable source) is greater than **x** (a chosen value)

The options are:

Function: **a>x**

V1: the source to be used as **a**

V2: the value for **x**

Using Thr as V1 (the source) and 0 as V2 (the trigger value) means that the switch would be "true" when the throttle was greater than it's midpoint (returned value =0) because sticks return values of between -100 to +100.



a<x

This is used to check if the value of **a** (a selectable source) is smaller (less) than **x** (a chosen value).

The options are:

Function: **a<x**

V1: the source to be used as **a**

V2: the value for **x**

Using Thr as V1 and 0 as V2 means that the switch would be true when the throttle stick was less than midpoint (returned value =0). Note that this differs from "a>x" in that this function is true when the compared value is less than whatever you've selected as x (V2) rather than when it is greater than the returned value of x.



|a|>x

This is used to check if the absolute value (meaning irrespective of + or -) of **a** (V1) is greater than **x** a chosen value (V2).

When using this switch, the options are:

Function: **|a|>x**

V1: the source to be used as **a**

V2: the value for **x**

Let's use Ail as V1 and 50 as V2. The Ail stick returns a value of 0 at it's middle point. The switch would be true (ON) if the stick position went past the halfway point either left or right (-50 or +50). It would be "false" (OFF) if the stick was returning an output between -50 and +50.



|a|<x

This function operates similar to "|a|>x" except that it is "true" when |a| is *less* than x where V1 is the selected source and x is the value against which you compare the value returned by V1. Because it's an absolute function, whether the returned value is a positive or negative value doesn't matter.

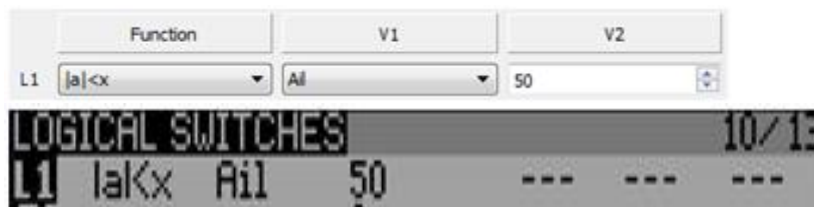
The options are:

Function: **|a|<x**

V1: the source to be used as **a**

V2: the value for **x**

Using Ail as V1 and 50 as V2 as we did above, the switch would be "true" (ON) when the Ail stick returned a value between -50 and +50, whereas with "|a|>x" it was "false" with the same condition.



AND

This switch checks that **BOTH** the switches selected in V1 **AND** V2 are true. If both switches are "true" (ON), then the logical switch is ON.

The options are:

Function: **AND**

V1: a switch (including logical and flight modes)

V2: a switch (including logical and flight modes)

Using SA↓ as V1 and SB↓ as V2, the switch would be "true" (ON) only if *both* SA↓ AND SB↓ were set. If SA ↓ was landing mode and SB↓ opened your gear doors, the logical switch could be used to check to see that both conditions were set before your landing gear actually deployed.



OR

This switch checks if *either* of the switches selected in V1 AND V2 are true.

The options are:

Function: **OR**

V1: a switch (including logical and flight modes)

V2: a switch (including logical and flight modes)

For example using SA↓ as V1 and SB↓ as V2, The switch could now be thought of as being true if SA or SB were down.



XOR (Exclusive or)

This switch checks if either, BUT NOT BOTH of the switches selected in V1 AND V2 are true

The options are:

Function: **XOR**

V1: a switch (including logical and flight modes)

V2: a switch (including logical and flight modes)

Using SA↓ as V1 and SB↓ as V2, the switch would be "true" (ON) if either SA or SB was down, but would return "false" (OFF) if both of them were down.



Edge

This is a momentary switch which can be activated by another switch (including logical and flight modes) and is dependent on the length of time that switch is "true". In this incidence, ON can indicate a physical position (low-med-high), another logical switch state, how long a flight mode is active, or even outputs from trim switches.

So when using this switch, the options are:

Function: **EDGE**

V1: an activating switch (including logical and flight modes)

V2: in two parts: first the minimum time the activating switch is required for, second the maximum period for the activating switch.

NOTE: If you leave duration as 0.0 for an EDGE switch, the switch will be on for ONLY 1 processing cycle (about 30ms)

There are three general settings for EDGE:

- *0.0 (release) -- the default -- the EDGE switch becomes active when the triggering switch is released.*
- (instant) -- The EDGE switch is ON once the triggering switch has been "true" for the minimum duration selected. This setting is activated by clicking the down arrow next to the default value of (released).
- The time value settings. See the examples below. They may seem complicated at first, but their use is where much of the power of the EDGE switch is configured. Take some time to understand them.

Some Examples:

Using SH↓ as V1 and 0.0 and (Instant) as V2: the logical switch would be true as soon as SH was pressed down.



Using SH↓ as V1 and 0.0 and --(Release) as V2: the logical switch would be true as soon as SH was pressed down AND then released



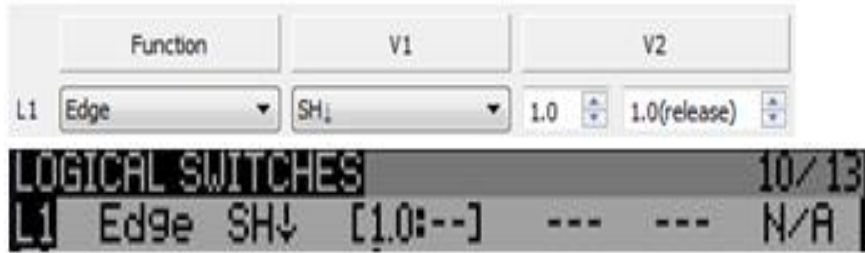
Using SH↓ as V1 and 0.0 and 1.0 as V2: the logical switch would be true as soon as SH was pressed down and released WITHIN 1 second. If it were held for longer than 1 second, this would not become true



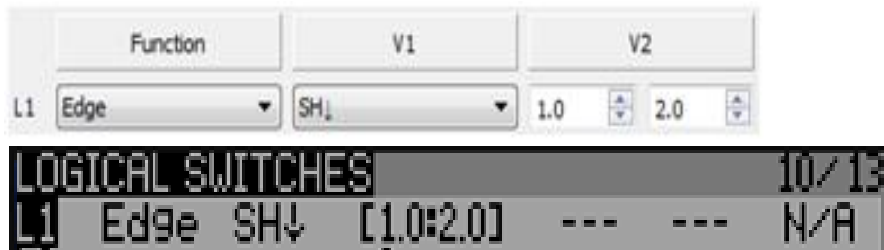
Using SH↓ as V1 and 1.0 and (Instant) as V2: the logical switch would be true as soon as SH had been held down for 1 second.



Using SH↓ as V1 and 1.0 and -1.0(release) as V2: the logical switch would be true as soon as SH was pressed down and held for 1 second (or more) AND then released.



Using SH↓ as V1 and 1.0 and 2.0 as V2: the logical switch would be true as soon as SH was pressed down, held for more than 1 second AND released before 2 seconds. If it were held for less than 1 second, or longer than 2 seconds, this would not become true.



a=b

This is used to check if the value of **a** (a selectable source) is equal to **b** (a different selectable source). It differs from "a=x" in that it compares two switch values directly without specifying numerical values for either.

The options are:

Function: **a=b**

V1: the source to be used as **a**

V2: the source to be used as **b**

For example using Ail as V1 and Rud as V2, the switch would be "true" (ON) when both the Ail and Rud outputs were equal (both at midpoint, etc.).



a>b

This is used to check if the value of **a** (a selectable source) is greater than **b** (a selectable source). See above for details.

The options are:

Function: a>b

V1: the source to be used as **a**

V2: the source to be used as **b**

For example using Ail as V1 and Rud as V2, the switch would be "true" (ON) if the Ail stick output was greater than the Rud stick output.

The switch would be read as: "Turn ON this logical switch when the Ail output is greater than the Rud output."



a<b

This is used to check if the value of **a** (a selectable source) is smaller than **b** (a selectable source). See **a>b** (above) for details. This function is true when the value of the first source as selected with V1 is greater than the returned value of the second source (V2).

The options are:

Function: $a < b$

V1: the source to be used as **a**

V2: the source to be used as **b**

For example using Ail as V1 and Rud as V2; means that the switch would be "true" (ON) when the value returned by the Ail stick was *less* than the value returned by the Rud stick.



$d \geq x$

This function compares a *change* in value to a set value. This is used to check if the Delta (the change in value) of **a** (a selectable source as chosen in V1) is greater than or equal to **x** as set in V2.

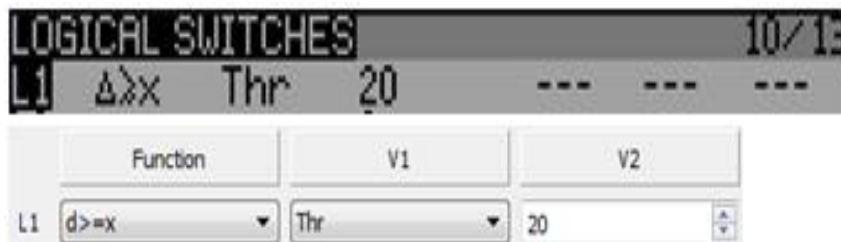
The options are:

Function: $d \geq x$

V1: the source to be used as **d**

V2: the value for **x**

For example using Thr as V1 and 20 as V2 the logical switch would be "true" (ON) if the output of the throttle stick was advanced by 20 units since the last time the switch was "true." If V2 had been set to -20, the switch would turn ON if the throttle value had decreased by 20 units



|d|>=x

This is used to check if the absolute (irrespective of + or -) Delta (the change since last true) value of **a** (a selectable source) is greater than or equal to **x** (a chosen value) as set in V2. It operates the same as "d>x" without the need to specify a positive or negative signed value.

The options are:

Function: d>=x

V1: the source to be used as **d**

V2: the value for **x**

For example using Thr as V1 and 20 as V2, the switch would be "true" (ON) if the throttle advanced or decreased by 20 or more points since the last time the switch was "true."



Timer

This function is used to turn a logical switch ON or OFF at specified intervals. This is a repeating on/off timer with both variable on and off times.

The options are:

Function: Timer

V1: the on time

V2: the off time

Using 1 as V1 and 0.5 as V2 gives a switch which is on for 1 second, then off for 0.5 seconds, and then turns on again for 1 second. This repeats continuously as long as the logical switch is true. Sometimes used in conjunction with AND to use another switch to activate the timer while the original timer function determines the operating parameters.



Sticky

This is a flip-flop. It can be thought of as another form of an on/off switch. It is turned on by the switch selected for V1 and turned off by the switch selected in V2.

The options are:

Function: $d \geq x$

V1: the switch to turn it on

V2: the switch to turn it off

Using SH↓ as V1 and SA↓ as V2, the logical switch will be ON when SH down is activated and the logical switch will turn off when SA is switched to down. Note that the item at V1 or V2 must **become** true to work. If SA is down, and you press SH down, the flip-flop will turn on. However, SA must be changed to SA↑, and then back to down in order to turn the flip-flop off.

When using Sticky, take care as the flip flop is turned on and off regardless of the AND switch. The logical switch will remain off if the AND Switch is false, but the flip-flop part will continue to turn on and off by V1 & V2, and is NOT reset by the AND switch.

