

Execution_signal example

The `execution_signal` parameter returns information about the specific signal that triggered execution. If there are multiple possible source signals, it identifies the one that initiated the execution. In this example, the Modbus RTU protocol is used. Configuration of devices:

name	description	device_alias	enable	protocol	id	device	baudrate	data bits	stopbits	parity	flowcontrol	scan_rate_ms	timeout_ms	serial_delay	execution_type	error_limit	queue_max	update
Elseta IOMod8DI8DO	IOMod 8DI8DO via modbus	IOMod88	1	Modbus rtu	1	PORT2	19200	8	1	even	none	3000	10000	150	signal	0	0	1
LUA device	LUA script	LUA	1	Lua runner														

`Execution_signal` should always be "signal" when `execution_signal` is used. PORT2 is used to connect IOMod 8DI8DO to WCC Lite via RS-485. IOMod signals configuration in this case:

signal_name	device_alias	signal_alias	source_device_alias	source_signal_alias	enable	math_expression	multiply	add	operation	units	min_value	max_value	absolute_threshold	threshold_units	suppression_time_ms	suppression_values	log	number_type	job_todo	tag_job_todo
DI1	IOMod88	DI1			1								0,1	real			1	DIGITAL	2;8;8	2;8;1
DI2	IOMod88	DI2			1								0,1	real			1	DIGITAL	2;8;8	2;9;1

In this case, IOMod's input signals are used, but other signals can be used depending on needs. Map the signals as shown below:

signal_name	device_alias	signal_alias	source_device_alias	source_signal_alias	threshold_units	absolute_threshold	enable	execute	tag_type
DI_1	LUA	DI1_lua	IOMod88	DI1	real	0	1	1	normal
DI_2	LUA	DI2_lua	IOMod88	DI2	real	0	1	1	normal
Results	LUA	results			real	0			normal

The first Lua signal is executing the first input signal, while the second Lua signal is executing the second input signal. The results signal is used to store information about the executed signal.

A Lua script can be written in various ways using the `execution_signal` parameter. However, in this example, a simple Lua script is used to retrieve the `signal_alias` of the executing signal by accessing `execution_signal.tag.signal_alias`:

```
local DI = execution_signal.tag.signal_alias

if DI == "DI1 lua" then
    publish(signals.results, 1)
elseif DI == "DI2 lua" then
    publish(signals.results, 2)
end
```

This is just one example of `execution_signal` usage. It can retrieve different information about executing a signal using:

- `execution_signal.tag.device_alias` - retrieves `device_alias`
- `execution_signal.tag.signal_alias` - retrieves `signal_alias`
- `execution_signal.value.value` - retrieves the value of the executing signal (for example, 1 or 0)
- `execution_signal.value.time` - retrieves system time in milliseconds (UNIX timestamp)
- `execution_signal.value.attributes` - retrieves attributes (for example, iv, nt, sb)

Upload the configuration to WCC Lite (it should upload without any errors):

PROTOCOL HUB	STATUS	SYSTEM	SERVICES	NETWORK	USERS
---------------------	--------	--------	----------	---------	-------

CONFIGURATION	IMPORTED SIGNALS	EVENT LOG	PROTOCOL CONNECTIONS	SCRIPT-RUNNER
----------------------	------------------	-----------	----------------------	---------------

Protocol configuration

IMPORT PROTOCOL CONFIGURATION

Here you can import Excel configuration file. Up to 1000 signals are allowed. All previous signals will be replaced.

Configuration file: 1 Choose File No file chosen 2 Import configuration

PLC (IEC-61499) Boot file: Choose File No file chosen Import FBOOT file

IEC61850 Client model file: Choose File No file chosen Import client model file

IEC61850 Server model file: Choose File No file chosen Import server model file

Upload the Lua script to the script runner and press start. After this, **Status** should show *Running*, and the script process number will appear.

PROTOCOL HUB	STATUS	SYSTEM	SERVICES	NETWORK	USERS	LOGOUT (ROOT)
---------------------	--------	--------	----------	---------	-------	---------------

CONFIGURATION	IMPORTED SIGNALS	EVENT LOG	PROTOCOL CONNECTIONS	SCRIPT-RUNNER
---------------	------------------	-----------	----------------------	----------------------

Script-Runner

LUA SCRIPT INSTANCE CONTROL

Script Configuration	Script process	Status	Script File
control1	-	Stopped	No Script provided 1 Upload Script Waiting for script

SAVED VALUE CLEARING

Clear all saved values

PROTOCOL HUB	STATUS	SYSTEM	SERVICES	NETWORK	USERS	LOGOUT (ROOT)
---------------------	--------	--------	----------	---------	-------	---------------

CONFIGURATION	IMPORTED SIGNALS	EVENT LOG	PROTOCOL CONNECTIONS	SCRIPT-RUNNER
---------------	------------------	-----------	----------------------	----------------------

Script-Runner

LUA SCRIPT INSTANCE CONTROL

Script Configuration	Script process	Status	Script File
LUA	-	Stopped	LUA.lua Upload Script Start 2

SAVED VALUE CLEARING

Clear all saved values

After activating the first input, in the WCC Lite web's imported signals tab, results should display 1:

PROTOCOL HUB

STATUS

SYSTEM

SERVICES

NETWORK

USERS

LOGOUT (ROOT)

WCC LITE

CONFIGURATION

IMPORTED SIGNALS

EVENT LOG

PROTOCOL CONNECTIONS

PROTOCOL LOGGER

SCRIPT-RUNNER

IMPORTED SIGNALS

☐ Column filter

Device name	Signal name	Device alias	Signal alias	Value	Units	State	Attributes	Time
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Elseta IOMod8DI8DO	DI1	IOMod88	DI1	1				2025-02-27 05:36:12.29
Elseta IOMod8DI8DO	DI2	IOMod88	DI2	0				2025-02-27 05:34:27.23
LUA device	DI_1	LUA	DI1_lua	1				2025-02-27 05:36:12.29
LUA device	DI_2	LUA	DI2_lua	0				2025-02-27 05:34:27.23
LUA device	Results	LUA	results	1				2025-02-27 05:36:12.31

After activating the second input, in the WCC Lite web's imported signals tab, results should display 2:

PROTOCOL HUB

STATUS

SYSTEM

SERVICES

NETWORK

USERS

LOGOUT (ROOT)

WCC LITE

CONFIGURATION

IMPORTED SIGNALS

EVENT LOG

PROTOCOL CONNECTIONS

PROTOCOL LOGGER

SCRIPT-RUNNER

IMPORTED SIGNALS

☐ Column filter

Device name	Signal name	Device alias	Signal alias	Value	Units	State	Attributes	Time
Elseta IOMod8DI8DO	DI1	IOMod88	DI1	0				2025-02-27 05:36:24.30
Elseta IOMod8DI8DO	DI2	IOMod88	DI2	1				2025-02-27 05:38:09.37
LUA device	DI_1	LUA	DI1_lua	0				2025-02-27 05:36:24.30
LUA device	DI_2	LUA	DI2_lua	1				2025-02-27 05:38:09.37
LUA device	Results	LUA	results	2				2025-02-27 05:38:09.39

Configuration --> Download

Lua script --> Download

🔄Revision #7

★Created 27 February 2025 08:12:22 by Simonas

✎Updated 27 May 2025 12:06:55 by Gabriele