Signal Delay with Lua

For delaying the response of command execution, Lua runner could be used as one of the solutions. In this example IEC104 and Modbus TCP are used. IEC104 protocol sends the command to Modbus and Lua signals and the results are represented as two separate signals. To create Excel configuration for WCC Lite in this case device sheet should look like this:

name	device_alias	enable	protocol	execution_type	error_limit	ip	id	mode	scan_rate_ms	retry_count	timeout_ms	bind_address	host	port	asdu_size	cot_size	ioa_size	rwt	swt	t1	t2	t3	time_sync	message_size	cache_size
Modbus TCP device	Modbus_TCP	1	Modbus TCP				1	tcp	300	3	10000	0.0.0.0		502											
IEC104 SCADA system	IEC104_SCADA	1	IEC 60870-5-104 slave									0.0.0.0		2404	2	2	3	8	12	15	10	20	1	249	100
LUA device	LUA	1	Lua runner	signal	0																				

In the fields marked red, for Modbus TCP enter IP address of Wi-Fi connected to computer and for IEC104 enter IP address of WCC Lite. Map the signals as shown below:

signal_name	device_alias	signal_alias	source_device_alias	source_signal_alias	execute	enable	gi	log	number_type	job_todo	tag_job_todo	common_address	info_address	data_type
Result_modbus	Modbus_TCP	result	LUA	result		1		1	SIGNED16	3;0;1	3;0;1			
command LUA	LUA	command	IEC104_SCADA	Command	1			1						
result LUA	LUA	result						1						
Command IEC104	IEC104_SCADA	Command	LUA	command		1	1	1				1	1	50

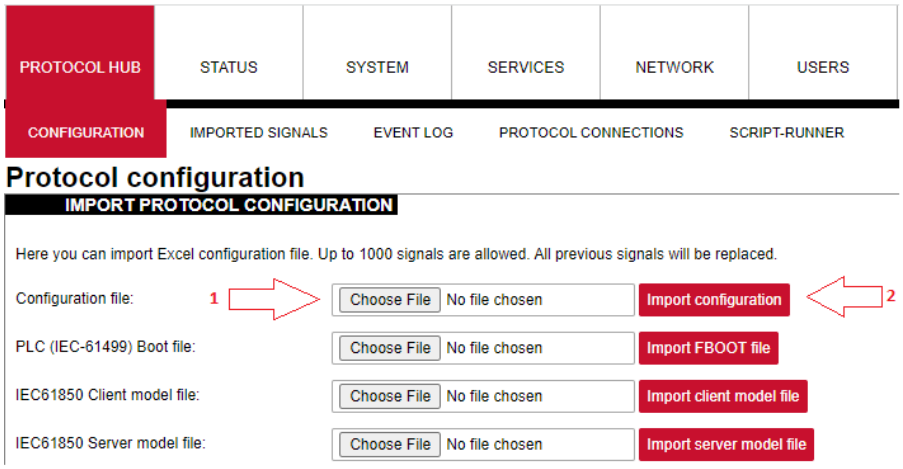
IEC104 SCADA will send command which will then go to Lua signal. Lua signal will send the response back to IEC104 SCADA and to Modbus TCP result signal.

For delaying signal response, Lua script could be written in many ways, however the most simple and effective one is to determine a wait time before publishing signal values to result signals. So in this case Lua script will look like this:

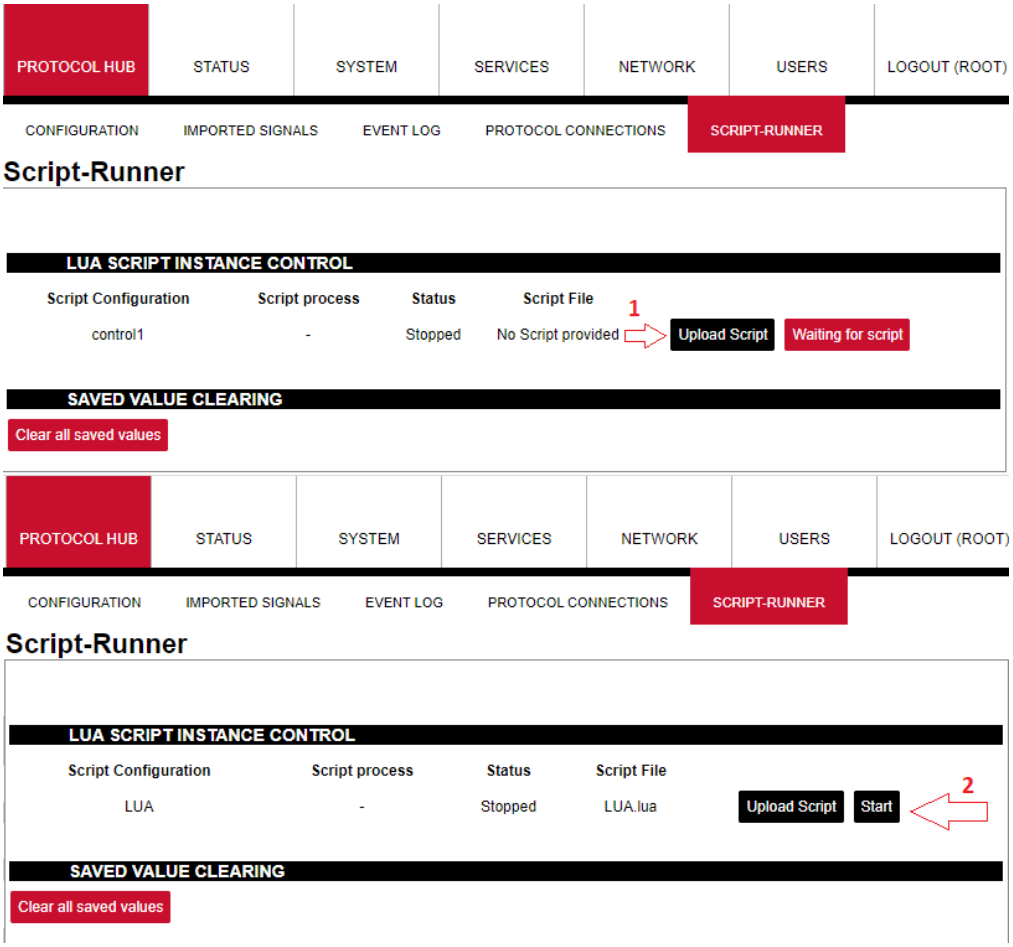
```
local saved = get(signals.result)--getting result signal which is equated to new variable 'saved'
local command = get(signals.command) --getting command signal which is equated to new variable 'command'

if not command then
    if saved then
        publish(signals.result, saved.value)--this value is published to result signals and saved value
    end
    return 0
end

if string.find(command.attributes, "nt") or string.find(command.attributes, "iv") or
string.find(command.attributes, "ov") then
   --searching if signal has negative attributes
  if command then
       command.attributes = "cot=7,cotn"--equates negative cot values to response signal attributes
        publish(signals.command, command)-- and publishes value to command signals and value
        if saved then
            publish(signals.result, saved.value)--restores saved value to result signals
        end
        return 0
    end
else
  if command then
        command.attributes = "cot=7"
        publish(signals.command, command)--in this cycle command value is being returned as well as
  --cot7 and cot10 values in case given signal is command type and has no negative attributes
        command.attributes = "cot=10"
        publish(signals.command, command) --publishes response to the command
        save(signals.result, command.value)--command value is being saved to result signal
    end
    local sleepTime = 30
    sleep(sleepTime) --before publishing values to result signals script waits 30s
    publish(signals.result, command.value)--in this row values are being published to result signals
```

```
end
```

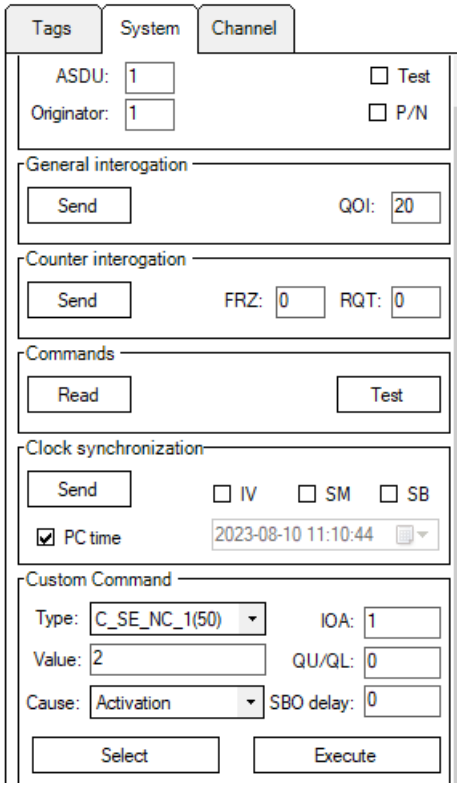
After entering values to empty Excel configuration fields, upload the configuration to WCC Lite (it should upload without any errors):



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



Open Vinci as IEC104 master, enter IP address of WCC Lite and start communication. Then open another Vinci window and connect Modbus TCP master – select Modbus TCP slave in Vinci and enter the same IP address as set in Excel configuration for Modbus device. With both communications running check **Protocol connections** on WCC Lite web interface, it should show *connected.* From IEC104 Vinci window go to *System* tab. Select command determined in the Excel configuration (50), IOA (1) and value (eg. 2).



Execute the command and check *Imported signals:*

IMPORTED SIGNALS

Device	Signal	Value	Units	State	Attributes	Time
Modbus TCP device	Result_modbus	0				2023-08-10 11:10:03.32
IEC104 SCADA system	Command IEC104	2		cmd	asdu=1,cot=6,ioa=1,org=1,ql=0,type=float	2023-08-10 11:35:40.18
LUA device	command LUA	2		cmd	asdu=1,cot=6,ioa=1,org=1,ql=0,type=float	2023-08-10 11:35:40.18
LUA device	result LUA	3				2023-08-10 10:04:14.94

After 30s result signals will now have the same value:

IMPORTED SIGNALS

Device	Signal	Value	Units	State	Attributes	Time
Modbus TCP device	Result_modbus	2				2023-08-10 11:36:10.52
IEC104 SCADA system	Command IEC104	2		cmd	asdu=1,cot=6,ioa=1,org=1,ql=0,type=float	2023-08-10 11:35:40.18
LUA device	command LUA	2		cmd	asdu=1,cot=6,ioa=1,org=1,ql=0,type=float	2023-08-10 11:35:40.18
LUA device	result LUA	2				2023-08-10 11:36:10.20

Signal delay could be used as a tool to synchronize signals so all the values are received at the same time. It can also be used to schedule commands or tasks when delay is required. Since Lua is one of the faster programming languages, it is the most effective instrument to be used is such matters.

Configuration --> Download

Lua script --> Download

Revision #9
Created 9 August 2023 08:58:34 by Gabriele
Updated 19 September 2023 06:38:36 by Gabriele