

Configuration Example With Default and Saved Values (Modbus TCP, IEC104)

Excel configuration

In order to control signals with Lua script an excel configuration for WCC Lite is needed. In this case three devices are required: one for Modbus TCP Master, one for IEC104-slave and one for Lua runner. Configuration example:

name	description	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	bo_size	nv	sv	t1	t2	t3	time_sync	message_size	cache_size
Modbus TCP device	Modbus TCP signals	Modbus_TCP	1	Modbus TCP				1	tcp	300	3	10000	0.0.0.0		502											
IEC104 SCADA system	IEC104 SCADA signals	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 signals	LUA	1	Lua runner	signal	0																				

In order to connect slave and master devices, master's IP address and slave's host addresses has to be specified. Modbus IP address will be the address of Wi-Fi to which the computer is connected (this can be checked on terminal window with command *ipconfig*) and host address for IEC104 slave protocol will be the IP address to which WCC Lite is connected. To reach the device via these addressed WCC Lite has to be connected to the internet.

When creating Excel configuration with Lua, there is an option **default value** for signal. This value will be set to the signal right after uploading configuration or if the script does not return any saved values.

Signals sheet:

signal_name	device_alias	signal_alias	source_device_alias	source_signal_alias	execute	default_value	threshold_units	absolute_threshold	enable	min_value	max_value	gi	log	number_type	job_todo	tag_job_todo	common_address	info_address	data_type
Result modbus	Modbus_TCP	result	LUA	result			real	2	1				1	SIGNED16	3;0;1	3;0;1			
command LUA	LUA	command	IEC104_SCADA	Command	1	20							1						
result LUA	LUA	result											1						
Command IEC104	IEC104_SCADA	Command	LUA	command					1	2	3	1	1				1	1	50

As it is seen from example values such as **min_value** and **max_value** can be added to determine limits of a signal. This way command signal will only return results which are within this range. Otherwise command value will have negative cot with invalid, non topical or overflow attributes and new value will not be sent to result signals. As configured, until command value is sent default value will be represented for Lua command signal. For saved values to be represented a Lua script is needed.

Lua script

Lua script example for this configuration is shown below:

```
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'

--get() function returns nill if there is no valid value

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

local time_diff = time_ms() - tonumber(command.time) --compares command time and real time
local is_command = time_diff < 30000 and time_diff > -30000 --if command time differs from
--real time more than 30s it will not be executed
```

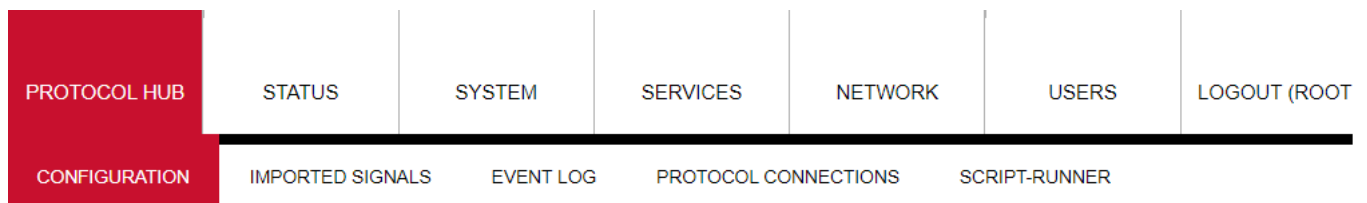
```

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 is_command then --if command execution time is not exceeding the limits 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 --if there is saved value then
            publish(signals.result, saved.value) --restores saved value to result signals
        end
        return 0
    end
end
else
    if is_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 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
    publish(signals.result, command.value) --in this row command value is being published to result signals
end
end

```

Uploading configuration and Lua script to WCC Lite



First Excel configuration needs to be uploaded to WCC Lite:



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:  No file chosen 

PLC (IEC-61499) Boot file: No file chosen

IEC61850 Server model file: No file chosen

After uploading configuration default value will be shown:

IMPORTED SIGNALS						
Device	Signal	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"/>
Modbus TCP device	Result_modbus					2023-08-22 16:53:40.45
IEC104 SCADA system	Command IEC104					
LUA device	command LUA	20				
LUA device	result LUA					

After uploading configuration no errors should appear and all signal should be represented on the web. To upload Lua script go to *Script-Runner*, select *upload script* and then *start*:

Script-Runner

LUA SCRIPT INSTANCE CONTROL

Script Configuration	Script process	Status	Script File	
CMD	-	Stopped	CMD.lua	<div>Upload Script</div> <div>Start</div>

SAVED VALUE CLEARING

Clear all saved values

Connecting master and slave via Vinci software

Connecting IEC104 slave

In order to connect to WCC Lite via IEC104 protocol, select Master(Client) mode on Vinci:

New

Protocol:

IEC 60870-5-104

Mode:

Master (Client)

Start

Check Settings tab to match excel configuration:

Settings

Console

Statistic

Structure

COT size in bytes:

2

ASDU size in bytes:

2

IOA size in bytes:

3

Parameters

☒ Send Start DT on start up

☒ Auto ack. Test Frame

Timeouts

t0 in seconds:

30

t1 in seconds:

45

t2 in seconds:

30

t3 in seconds:

200

Security

☐ Enable TLS

Windows

RWT (w) size:

8

SWT (k) size:

12

Specify IP address which should match the one in Excel configuration.

Connecting ModbusTCP master:

To connect ModbusTCP Master, select Slave (Server) mode on Vinci:

New

Protocol:

Modbus TCP

Mode:

Slave (Server)

Start

Check address to match **id** in Excel configuration:

Settings

Console

Statistic

Address

Select all

Clear all

☒

1

☐

2

☐

3

☐

4

☐

5

☐

6

☐

7

☐

8

☐

9

☐

10

☐

11

☐

12

☐

13

☐

14

Value

Default value: 0

Match the IP address given in Excel configuration as well.

Executing commands

Start both master and slave simulations on Vinci. Check if both protocols are connected to WCC Lite on the web tab *Protocol Connections*:

PROTOCOL CONNECTIONS				
Device	Protocol	Host	Status	Timestamp
Modbus_TCP	Modbus TCP master	192.168.67.176	Connected	2023-08-22 14:25:37
IEC104_SCADA	IEC 60870-5-104 slave	192.168.1.216	Connected	2023-08-22 14:25:11

To execute commands, open Vinci program with IEC104 master running. Here, go to *System* tab and fill in required fields such as IOA and select data type indicated in Excel configuration. First, try sending value that is outside the set range:

Custom Command

Type:

C_SE_NC_1(50)

IOA:

1

Value:

1

QU/QL:

0

Cause:

Activation

SBO delay:

0

Select

Execute

After selecting *execute* this value will not be showed on the web and positive cot6 and negative cot7 values will be seen on Vinci IEC104 simulation window. Positive cot6 indicates command activation and negative cot7 means that command activation confirmation was denied. The command signal value will not be represented as result signals, because it is determined in the script, that signals with negative attributes will not be published.

IMPORTED SIGNALS						
Device	Signal	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"/>
Modbus TCP device	Result_modbus					2023-08-25 12:32:03.43
IEC104 SCADA system	Command IEC104	2		coln	cot=7	2023-08-25 13:04:30.72
LUA device	command LUA	2		cmd,lv	asdu=1,cot=6,ioa=1,org=1,qi=0,type=float	2023-08-25 13:04:30.72
LUA device	result LUA					

Now specify value which will be sent as a result and is within the given range.

PROTOCOL HUB	STATUS	SYSTEM	SERVICES	NETWORK	USERS	LOGOUT (ROOT)	
CONFIGURATION	IMPORTED SIGNALS	EVENT LOG	PROTOCOL CONNECTIONS	SCRIPT-RUNNER			

IMPORTED SIGNALS						
Device	Signal	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"/>
Modbus TCP device	Result_modbus	2				2023-08-07 12:18:52.66
IEC104 SCADA system	Command IEC104	2			cot=10	2023-08-07 12:17:57.42
LUA device	command LUA	2		cmd	asdu=1,cot=6,ioa=1,org=1,qi=0,type=float	2023-08-07 12:17:57.42
LUA device	result LUA	2				2023-08-07 12:18:52.66

Positive cot7 and cot10 values will be seen on the Vinci IEC104 simulation window:

Settings	Console	Statistic										
Time	Source	Message	TI	Cause	ASDU	IOA	Value	Status			RawData	
09:18:16.942	192.168.73.216:2404	RSN:3 SSN:40	C_SE_NC_1 (50)	Pos. ActCon (7) (T=0 O=0)	1	1	2	Execute Default			68 12 50 00 06 00 32 01 07 00 01 00 01 00 00 00 00 40 00	
09:18:16.993	192.168.73.216:2404	RSN:3 SSN:41	C_SE_NC_1 (50)	Pos. ActTerm (10) (T=0 O=0)	1	1	2	Execute Default			68 12 52 00 06 00 32 01 0A 00 01 00 01 00 00 00 00 40 00	
09:18:17.004	Vinci	RSN:42									68 04 01 00 54 00	
09:18:17.935	192.168.73.216:2404	RSN:3 SSN:42	C_SE_NC_1 (50)	Pos. ActCon (7) (T=0 O=0)	1	1	2	Execute Default			68 12 54 00 06 00 32 01 07 00 01 00 01 00 00 00 00 40 00	
09:18:17.985	192.168.73.216:2404	RSN:3 SSN:43	C_SE_NC_1 (50)	Pos. ActTerm (10) (T=0 O=0)	1	1	2	Execute Default			68 12 56 00 06 00 32 01 0A 00 01 00 01 00 00 00 00 40 00	
09:18:18.933	192.168.73.216:2404	RSN:3 SSN:44	C_SE_NC_1 (50)	Pos. ActCon (7) (T=0 O=0)	1	1	2	Execute Default			68 12 58 00 06 00 32 01 07 00 01 00 01 00 00 00 00 40 00	
09:18:18.982	192.168.73.216:2404	RSN:3 SSN:45	C_SE_NC_1 (50)	Pos. ActTerm (10) (T=0 O=0)	1	1	2	Execute Default			68 12 5A 00 06 00 32 01 0A 00 01 00 01 00 00 00 00 40 00	
09:18:19.942	192.168.73.216:2404	RSN:3 SSN:46	C_SE_NC_1 (50)	Pos. ActCon (7) (T=0 O=0)	1	1	2	Execute Default			68 12 5C 00 06 00 32 01 07 00 01 00 01 00 00 00 00 40 00	
09:18:19.993	192.168.73.216:2404	RSN:3 SSN:47	C_SE_NC_1 (50)	Pos. ActTerm (10) (T=0 O=0)	1	1	2	Execute Default			68 12 5E 00 06 00 32 01 0A 00 01 00 01 00 00 00 00 40 00	
09:18:20.936	192.168.73.216:2404	RSN:3 SSN:48	C_SE_NC_1 (50)	Pos. ActCon (7) (T=0 O=0)	1	1	2	Execute Default			68 12 60 00 06 00 32 01 07 00 01 00 01 00 00 00 00 40 00	
09:18:21.001	192.168.73.216:2404	RSN:3 SSN:49	C_SE_NC_1 (50)	Pos. ActTerm (10) (T=0 O=0)	1	1	2	Execute Default			68 12 62 00 06 00 32 01 0A 00 01 00 01 00 00 00 00 40 00	

This value will also be represented on Modbus TCP master Vinci simulation window:

12:21:36.324	192.168.67.133:5...	1	3	0	1	09 13 00 00 00 06 01 03 00 00 00 01
12:21:36.344	VINCI	1	3			00 02 09 13 00 00 00 05 01 03 02 00 02
12:21:36.648	192.168.67.133:5...	1	3	0	1	09 14 00 00 00 06 01 03 00 00 00 01
12:21:36.728	VINCI	1	3			00 02 09 14 00 00 00 05 01 03 02 00 02

To show what happens if the value is not within determined range after the correct value has been sent before, try executing command with smaller or larger value specified:

09:22:40.066	Vinci	RSN:62 SSN:3	C_SE_NC_1 (50)	Pos. Act (6) (T=0 O=1)	1	1	1	Execute Default	68 12 06 00 7C 00 32 01 06 01 01 00 01 00 00 00 00 80 3F 00
09:22:41.071	192.168.73.216:2404	RSN:4 SSN:62	C_SE_NC_1 (50)	Neg. ActCon (7) (T=0 O=0)	1	1	2	Execute Default	68 12 7C 00 08 00 32 01 47 00 01 00 01 00 00 00 00 40 00
09:22:42.090	192.168.73.216:2404	RSN:4 SSN:63	C_SE_NC_1 (50)	Neg. ActCon (7) (T=0 O=0)	1	1	2	Execute Default	68 12 7E 00 08 00 32 01 47 00 01 00 01 00 00 00 00 40 00
09:22:43.071	192.168.73.216:2404	RSN:4 SSN:64	C_SE_NC_1 (50)	Neg. ActCon (7) (T=0 O=0)	1	1	2	Execute Default	68 12 80 00 08 00 32 01 47 00 01 00 01 00 00 00 00 40 00
09:22:44.079	192.168.73.216:2404	RSN:4 SSN:65	C_SE_NC_1 (50)	Neg. ActCon (7) (T=0 O=0)	1	1	2	Execute Default	68 12 82 00 08 00 32 01 47 00 01 00 01 00 00 00 00 40 00

Again, positive cot6 and negative cot7 values is seen on Vinci window.

PROTOCOL HUB	STATUS	SYSTEM	SERVICES	NETWORK	USERS	LOGOUT (ROOT)	
CONFIGURATION	IMPORTED SIGNALS	EVENT LOG	PROTOCOL CONNECTIONS	SCRIPT-RUNNER			

IMPORTED SIGNALS						
Device	Signal	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"/>
Modbus TCP device	Result_modbus	2				2023-08-07 12:23:07.22
IEC104 SCADA system	Command IEC104	2		coln	cot=7	2023-08-07 12:22:40.21
LUA device	command LUA	2		cmd,lv	asdu=1,cot=6,ioa=1,org=1,qi=0,type=float	2023-08-07 12:22:40.21
LUA device	result LUA	2				2023-08-07 12:23:07.22

As seen on WCC Lite web window, command signals have negative attributes and result signals have the same value as before, because it was saved by Lua script. This example shows that Lua runner can be used to save certain values to signal. For example after the restart saved value could be seen on command signal to determine minimum or maximum value, last value or typical value. This solution could be useful for protecting important data even after reboot or connection faults.

Configuration --> [Download](#)

Lua script --> [Download](#)

🕒Revision #13

★Created 15 December 2022 08:21:57 by Gabriele

✎Updated 19 September 2023 06:38:30 by Gabriele