

Cronjob example

Lua device can be configured using cron time expression. This way script execution can be scheduled or executed at certain times. To create such solution, device sheet of Excel configuration should look like this:

name	description	device_alias	enable	protocol	execution_type	execution_parameter	error_limit	ip	id	mode	scan_rate_ms	retry_count	timeout_ms	bind_address	host	port	asdu_size	cot_size	loa_size	rwt	swt	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	date	*/30 * * * *	0																				

As seen in this configuration, execution type for Lua device is date and execution parameter is in cron time expression. In this case script will be executed every 30s starting from a mm:30 or a mm:00 mark. There are a lot of online cron expression parsers or generators to convert this expression to a more understandable form: <https://crontab.cronhub.io/>.

A Cron expression must have six variables, for instance, the code "0 * * * *" will not suffice because it contains only five variables. To rectify this, add a "0" to the beginning of the code: "0 0 * * * *". Similarly, the code "0 20-23,0-4,11 * * *" which is displayed as correct on the website, to achieve the effect of every hour from 08:00 PM to 11:59 PM, 12:00 AM to 04:59 AM, and 11:00 AM, it must be adjusted as follows: "0 0 20-23,0-4,11 * * *".

To complete Excel configuration fill out the red fields with correct parameters. For "ip" field enter IP address of Wi-Fi connection that is connected to computer. This can be checked by entering command "ipconfig" on terminal window. For the "host" field enter IP address of WCC Lite device.

Signals sheet should look similar to this:

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	FLOAT	3;0;1	3;0;1			
command LUA	LUA	command	IEC104_SCADA	Command				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.

Lua script in this case will be very simple. To show how Lua with cronjob can be used in real life, a calculation for kilowatts per hour has been added:

```
local kW = tonumber(get_value(signals.command))--function "get_value" will get value from iec104 command
--without any attributes this value is still in string form so funtion "tonumber" will convert it to number
--a new variable is creted which is now equal to command value
value = value or 0 --new variable is created. It has to be equal to itself or to 0 if the script is running
--for the first time
value = (kW * 30/3600) + value --formula for calculating kilowatts. command value from iec104 is multiplied by
time.
--Since the script is being executed every 30s, this time needs to be converted to hours. an old value is being
--added to new value, this way result value that is being published will grow every 30s
publish(signals.result, value)--publishes result to result signals for lua and modbus TCP.
--Since the script is being executed every 30s, those values will be refreshed every 30s as well
```

To test the functionality of this script upload Excel 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

2

PLC (IEC-61499) Boot file:

IEC61850 Client model file:

IEC61850 Server model file:

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

SAVED VALUE CLEARING

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

2

SAVED VALUE CLEARING

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 (for example 3600).

Tags

System

Channel

Commands

Read

Test

Clock synchronization

Send

☐ IV
☐ SM
☐ SB

☒ PC time

2023-08-17 09:14:03

Custom Command

Type: C_SE_NC_1(50)

IOA: 1

Value: 3600

QU/QL: 0

Cause: Activation

SBO delay: 0

Select

Execute

Execute the command and check *Imported signals*:

IMPORTED SIGNALS						
Device	Signal	Value	Units	State	Attributes	Time
Modbus TCP device	Result_modbus	0				2023-08-17 10:19:00.86
IEC104 SCADA system	Command IEC104	3600		cmd	asdu=1,cot=6,ioa=1,org=1,qi=0,type=float	2023-08-17 10:19:05.20
LUA device	command LUA	3600		cmd	asdu=1,cot=6,ioa=1,org=1,qi=0,type=float	2023-08-17 10:19:05.20
LUA device	result LUA					

On a 30s mark result signals will have calculated value:

IMPORTED SIGNALS						
Device	Signal	Value	Units	State	Attributes	Time
Modbus TCP device	Result_modbus	30				2023-08-17 10:19:30.35
IEC104 SCADA system	Command IEC104	3600		cmd	asdu=1,cot=6,ioa=1,org=1,qi=0,type=float	2023-08-17 10:19:05.20
LUA device	command LUA	3600		cmd	asdu=1,cot=6,ioa=1,org=1,qi=0,type=float	2023-08-17 10:19:05.20
LUA device	result LUA	30				2023-08-17 10:19:30.30

After another 30s new value will be added to old value and result signals will be updated accordingly:

IMPORTED SIGNALS						
Device	Signal	Value	Units	State	Attributes	Time
Modbus TCP device	Result_modbus	60				2023-08-17 10:20:00.36
IEC104 SCADA system	Command IEC104	3600		cmd	asdu=1,cot=6,ioa=1,org=1,qi=0,type=float	2023-08-17 10:19:05.20
LUA device	command LUA	3600		cmd	asdu=1,cot=6,ioa=1,org=1,qi=0,type=float	2023-08-17 10:19:05.20
LUA device	result LUA	60				2023-08-17 10:20:00.39

Cronjob with Lua can be used mainly to schedule tasks. It can also be used as a way to filter, monitor or control the data. As seen in this example, Lua script can help calculate certain parameters which will then can be sorted using cron time expression.

Configuration --> [Download](#)

Lua script --> [Download](#)