


13.3 Modbus Slave

WCC Lite can act as one (or several) of slave devices in a communication line. This can be used to transmit data to SCADA systems or other RTU devices. It can reply to messages from Modbus Master with matching devices and register addresses.

Configuring datapoints

Modbus Slave in WCC Lite has to be configured via Excel. This configuration contains two Excel sheets where parameters have to be filled in - Devices and Signals

 If TCP/IP is used as a transmission medium, only devices with IPs predefined in the host column are allowed to connect. All other connections are rejected

Modbus Slave parameters for Devices tab

Parameter	Type	Description	Required		Default Value (when not specified)	Range	
			TCP	RTU/ASCII		Min	Max
name	string	User-friendly name for a device	Yes	Yes			
description	string	Description of a device	No	No			
device_alias	string	Alphanumeric string to identify a device	Yes	Yes	unknown		
enable	boolean	Enabling/disabling of a device	No	No	1	0	1
protocol	string	Protocol to be used	Yes	Yes		Modbus serial Slave, Modbus TCP Slave	
host	string	Space-separated host IP addresses of master device	Yes	-			
port	integer	TCP port to listen for incoming connections	Yes	-			
bind_address	string	The IP address of the network adapter used to connect to the slave device (Default: "0.0.0.0")	No	No	0.0.0.0		
keep_alive_timeout	integer	The minimum time a connection can be idle without being closed in milliseconds	No	No	60		
mode	string	Choosing between RTU ("rtu"), ASCII ("ascii") and TCP("tcp") modes. ASCII is the same as RTU, but with ASCII symbols.	No	No	TCP (for TCP) RTU (for Serial)	rtu, ascii, tcp	
device	string	Communication port ("PORT1"/"PORT2")	-	Yes		PORT1	PORT2
baudrate	integer	Communication speed, baud/s	-	Yes	9600	300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200	
databits	integer	Data bit count for communication	-	Yes	8	6	9
stopbits	integer	Stop bit count for communication	-	Yes	1	1	2

parity	string	Communication parity option	-	Yes	none	none, even, odd
flowcontrol	string	The communication device's flow control option.	-	No	none	none

Modbus Slave parameters for the Signals tab

Parameter	Type	Description	Required		Default Value (when not specified)	Range	
			TCP	RTU/A SCII		Min	Max
signal_name	string	User-friendly signal name	Yes	Yes			
device_alias	string	Alphanumeric string to identify a device	Yes	Yes			
signal_alias	string	Unique alphanumeric name of the signal to be used	Yes	Yes			
enable	boolean	Enabling/disabling an individual signal	No	No	1	0	1
number_type	string	Type of a number (FLOAT, DOUBLE, DIGITAL, etc.). This defines the size that will be read.	Yes	Yes			
log	integer	If the log parameter is equal to 1, the signal values and attributes will be seen in the events log.	No	No	0		
slave_id	integer	Address of a slave device	Yes	Yes			
function	integer	Function number	Yes	Yes		1, 2, 3, 4	
register_addresses	integer	Register address	Yes	Yes			
periodic_update_ms	integer	Signal value is published periodically according to the value set.	No	No	-	-	-


Device status signals

Modbus slave has an additional signal which can be configured to show communication status. It indicates if the master device has disconnected from the slave (WCC Lite). To configure such signal for Modbus protocol, `job_todo` and `tag_job_todo` fields with string values are required. For Modbus slave required parameters for the status, the signal will be **signal_name**, **device_alias**, **signal_alias**, **number_type**, **slave_id**, **function**, **register_address**, **job_todo** and **tag_job_todo**. `job_todo` value must be *device_status* and for `tag_job_todo` there are 4 variations: `communication_status`, `device_running`, `device_error`, `unknown_error`. Each signal has 4 possible values and is based on the same logic. If the signal returns the value of 0, it means an unknown error has appeared, 1 – device or protocol connection is on and working properly, 2 – device is off or protocol is disconnected, 3 – error or service is down.

Mapping values to registers

Internally stored values aren't organised in a register-like order, therefore mapping should be done by the user. This mapping includes setting the address of the device WCC Lite is simulating as well as the function number, register number and how many 16-bit registers are used to store a value. These values should be set in `common_address`, `function`, `info_address` and `size` columns respectively in the Excel configuration.

To find out how many registers should be used for storing values, and how values can have their values swapped, a user should consult a section [number_type](#).

 If a Modbus master device requests data from a register that is mapped but doesn't yet have an initial value, an **ILLEGAL DATA ADDRESS** error code will be returned. The same error code is returned if the requested size of the value is bigger than defined or if the register is not configured at all.

Debugging a Modbus Slave application


If the configuration for Modbus Slave is set up, a handler for the protocol will start automatically. If the configuration is missing or contains errors, the protocol will not start. It is done intentionally to decrease unnecessary memory usage.


Modbus Slave command line debugging options

To run a debug session:

- **Step 1:** Service must be stopped by entering the following command into the wccliite:
/etc/init.d/modbus-slave stop
- **Step 2:** After the service is stopped it must be started with the preferred configuration file (JSON files found in the /etc/ folder) and a debug level 7: **modbus-slave -c /etc/modbus-slave/modbus-slave.json -d7**
Additional output forming options described in the table below.
- **Step 3:** Once the problem is diagnosed normal operations can be resumed with the following command: **/etc/init.d/modbus-slave start**

```
-h [ -help ] Display help information
-V [ -version ] Show version
-d<debug level> Set debugging level
-c [ -config ] Config path
-e [ -redis ] Show redis debug information
```

 If Modbus Slave does not work properly (e.g. no communication between devices, data is corrupted, etc.), a user can launch a debug session from the command line interface and find out why the link is not functioning properly.

 To launch a debugging session, a user should stop `modbus-slave` process and run `modbus-slave` command with respective flags as shown above.

🔄Revision #3

★Created 22 November 2024 07:15:20 by Gabriele

✎Updated 13 December 2024 11:00:15 by Gabriele