Modbus registers – BDVA Oil-Stick TS 4

Rev. 1.1 from March 26, 2020

Modbus is available over TCP and UDP. In both cases it's on port 502.

Implementation of Modbus server supports following functions:

- READ_COILS (function code 1)
- READ_DISCRETE_INPUTS (function code 2)
- READ_MULTIPLE_HOLDING_REGISTERS (function code 3)
- READ_INPUT_REGISTERS (function code 4)
- WRITE_SINGLE_COIL (function code 5)
- WRITE_SINGLE_HOLDING_REGISTER (function code 6)
- WRITE_MULTIPLE_COILS (function code 15)
- WRITE_MULTIPLE_HOLDING_REGISTERS (function code 16)

All values are big-endian.

Reading or editing service register may cause improper working of device. In some cases, device can get damaged.

Coils Registers

name	type	address	description
StartMeasurement	bool	00000	1 - start measurement, 0 - stop measurement
PauseMeasurement	bool	00001	1 - pause measurement, 0 - resume measurement
AutostartMeasurement	bool	00002	if set to 1, StartMeasurement coil will be set to 1 whenever device is powered on, default set to 1
SERVICE	N/A	00003 – 00011	Service Register – Do Not Read / Write or Edit
SetIP	bool	00012	* sets IP address of the device to value stored in `IP` holding register
SetNetmask	bool	00013	* sets netmask of the device to value stored in `Netmask` holding register
SetGateway	bool	00014	* sets gateway address of the device to value stored in `Gateway` holding register
SERVICE	N/A	00015	Service Register – Do Not Read / Write or Edit

* requires device restart to take effect

Discrete Registers

name	type	address	description
SERVICE	N/A	10000 – 10007	Service Register – Do Not Read / Write or Edit

Input Registers

name	type	address	description
SerialNumber	uint8_t[16]	30000	serial number of the device
SoftwareVersion	uint8_t[4]	30008	array storing version info in following order { 0, major, minor, fix }
BuildDateTime	uint8_t[20]	30010	string containing date and time at which software was built
MeasurementCount	uint32_t	30020	number of last measurements
Status	uint16_t	30022	status flags (status codes are listed below this table)
TM	uint16_t	30023	* temperature in °C
AVG_RS	uint16_t	30024	* average humidity from 333 samples
BDV	uint16_t	30025	* BDV for 20°C (equivalent of IEC 60156)
BDVCR	uint16_t	30026	* BDV for current Temperature
WC	uint16_t	30027	* water content
AVG_WC_S	uint16_t	30028	* average WC from 333 samples (~5.5 minutes assuming 1MPS[#1])
AVG_WC_M	uint16_t	30029	* average WC from 999 samples (~16.7 minutes assuming 1MPS)
AVG_WC_L	uint16_t	30030	* average WC from 2997 samples (~50 minutes assuming 1MPS)
AVG_BDV_S	uint16_t	30031	* average BDV from 333 samples (~5.5 minutes assuming 1MPS)
AVG_BDV_M	uint16_t	30032	* average BDV from 999 samples (~16.7 minutes assuming 1MPS)
AVG_BDV_L	uint16_t	30033	* average BDV from 2997 samples (~50 minutes assuming 1MPS)
AVG_BDVCR_S	uint16_t	30034	* average BDVCR from 333 samples (~5.5 minutes assuming 1MPS)
AVG_BDVCR_M	uint16_t	30035	* average BDVCR from 999 samples (~16.7 minutes assuming 1MPS)
AVG_BDVCR_L	uint16_t	30036	* average BDVCR from 2997 samples (~50 minutes assuming 1MPS)
SERVICE	N/A	30037 – 30192	Service Register – Do Not Read / Write or Edit

* fixed point value with 2 decimals
([#1]: MPS – means Measurement Per Second)

Status codes

Value	Description
0x0000	measurement was successful
0x0002	buffer 1 is not full, some calculations were omitted
0x0004	buffer 2 is not full, some calculations were omitted
0x0800	triangulation has failed, some calculations were omitted

Holding Registers

name	type	address	description
IP	uint32_t	40000	user defined IP address
Netmask	uint32_t	40002	user defined network mask
Gateway	uint32_t	40004	user defined gateway address
SERVICE	N/A	40005 – 40054	Service Register – Do Not Read / Write or Edit

Network properties

During first launch of the device default network properties will be used. From that point user can define desired network properties by following steps in `Change network properties` section.

Default network properties

```
address 192.168.0.10
netmask 255.255.255.0
gateway 192.168.0.1
```

Change network properties

Network properties can be modified via Modbus server.

To change any of the network properties, they need to be set in following holding registers:

- `IP`,
- `Netmask`,
- `Gateway`,

and then saved to EEPROM memory with corresponding coils:

- `SetIP`,
- `SetNetmask`,
- `SetGateway`.

Note: Those coils will be automatically reset to 0 after values get saved to EEPROM memory.

Example

To change network properties to

```
address 2.3.4.5
netmask 255.254.0.0
gateway 2.3.4.1
```

1. Set 2 holding registers starting at address `IP` to value **0x0203** at address **40000** and value **0x0405** at address **40001**.
2. set 2 holding registers starting at address `Netmask` to value **0xFFFE** at address **40002** and value **0x0000** at address **40003**.
3. set 2 holding registers starting at address `Gateway` to value **0x0203** at address **40004** and value **0x0401** at address **40005**.
4. Set coil at address `SetIP` to 1.
5. Set coil at address `SetNetmask` to 1.
6. Set coil at address `SetGateway` to 1.
7. Restart device (after failsafe period, explained next) for changes to take effect.
8. Connect to the modbus TCP server on a new address so that the new properties would be saved to EEPROM. Otherwise failsafe mechanism (described below) will trigger.

Failsafe

If, after changing IP address and restarting device, no successful TCP connection to the modbus server has been established, the old network properties will be restored on next device restart.

Measurements

Measurements can be started by setting `StartMeasurement` coil to 1.

Pause

To pause measurement set `PauseMeasurement` coil to 1. To resume measurement `PauseMeasurement` set it back to 0.

Restart

To restart measurement set `StartMeasurement` coil to 0. Wait 2 seconds. Set `StartMeasurement` coil to 1.

Autostart

To enable autostart of measurement on power on set `AutostartMeasurement` coil to 1. After restart of the device both `AutostartMeasurement` and `StartMeasurement` coils should automatically be set to 1 and measurement should be in progress.

Advertising

Every 10 seconds `UDP` packet will be sent out without any target. This packet contains:

- header, 4 bytes, a string `"NET-"`
- serial, 16 bytes, serial of the device *(last bytes might not be part of the serial number)*
- IP address, 4 bytes,
- network mask, 4 bytes,
- gateway address, 4 bytes,

Example

For device with following properties:

- serial: `SERIAL0123456`
- IP: 192.168.0.10
- netmask: 255.255.255.0
- gateway 192.168.0.1

generated advertisement message would look like this:

```
4E45542D53455249414c303132333435362020C0A8000AFFFFFF00C0A80001
```

where:

- `0x4E 0x45 0x54 0x2D` - header
- `0x53 0x45 0x52 0x49 0x41 0x4C 0x30 0x31 0x32 0x33 0x34 0x35 0x36 0x20 0x20 0x20` - serial
- `0xC0 0xA8 0x00 0x0A` - IP address
- `0xFF 0xFF 0xFF 0x00` - netmask
- `0xC0 0xA8 0x00 0x01` - gateway address

*Table **Fehler! Kein Text mit angegebener Formatvorlage im Dokument.**-1 Notes*

Date	Author	Description of changes

Date	Author	Description of changes

page intentionally left blank