
SNMP

NETIO M2M API protocols docs

Protocol version: **NETIO-MIB 1.05**

Short summary

SNMP is implemented as M2M API protocol, where NETIO device can use SNMP protocol allowing monitoring and control of NETIO power outputs (power sockets 230V or power outlets IEC-320 110/230V).

- NETIO 4x devices (NETIO 4 / 4All / 4C) supports following SNMP versions:
 - **v1,2c - SNMP v1 / v2c** are allowing only status monitoring (**Read only**)
 - **v3 - SNMP v3** is using authorization and encryption therefore allow monitoring and control each output (**Read & Write**).
- Used with NETIO 4All, the protocol also includes power consumption values.
- SNMP M2M API protocol must be enabled first in the WEB configuration of the respective device. For details, see the “NETIO WEB configuration” chapter.

Supported devices

- NETIO 4All
- NETIO 4 (Energy metering not supported)
- NETIO 4C (Energy metering not supported)

*Note: **NETIO 4x** means all NETIO 4 devices (NETIO 4 / 4All / 4C)*

Supported devices and firmware

NETIO 4x firmware – 3.1.0 and later

NOTE: *This document provides basic info about the M2M API protocol. Other device functions are described in the product manual.*

Quick start with SNMP & NETIO

NETIO MIB is available for download at the web administration – “M2M API Protocols” / “SNMP” tab.

Following standard MIBs are also required:

- SNMPv2-SMI
- SNMPv2-TC

SNMP v3 – control of outputs

Object <i>OID</i> x – output number (1, 2, 3, 4)	Type	Value	Action
netioOutputAction.x 1.3.6.1.4.1.47952.1.1.1.5.x	INTEGER (i)	0	Turn OFF
		1	Turn ON
		2	Short OFF delay (restart)
		3	Short ON delay
		4	Toggle (invert the state)
		5	No change

Example:

```
snmpset -m NETIO-PRODUCTS-NETIO-MIB -M /usr/share/snmp/mibs/ -v 3 -a SHA -A a1234567 -l  
authPriv -u admin -x AES -X a1234567 192.168.2.78 netioOutputAction.1 i 1
```

General NETIO 4x output functions

Output status – “read” function

- **0** – Power **OFF**
- **1** – Power **ON**

Output actions – “write” function

- **0** – Turn **OFF**
- **1** – Turn **ON**
- **2** – Short OFF delay (restart)
- **3** – Short ON delay
- **4** – Toggle (invert the state)
- **5** – No change

Short ON / OFF delay

This command switches a power output On / Off for a defined time. It is useful for example to power-cycle a server with a defined switch-off time, or to switch on a pump for a defined time.

This “short” delay is protected: the power output will remain in the defined state regardless of any other M2M requests received. During this time, the output state can only be changed by pressing the button on the NETIO device and this action cancel M2M short ON/OFF command for the particular output. Other requests to control the particular output are simply ignored.

The short ON / OFF delay interval can be defined in the device web administration. It is specified in ms (milliseconds) and rounded up to hundreds of milliseconds (0,1s).

This interval can be also defined using some M2M API protocol commands. In that case, it is valid only for a single protocol session (the following short ON / Short OFF command). When the connection is closed or restarted, the interval is reset to the device default value (defined in the web administration for each output).

Security issues

Do not use default usernames and passwords! Keep your Ethernet and WiFi networks secured.

Power-Up outputs state

All outputs are Off during the first 25 to 30 seconds after power-up.

After this time, all outputs are set to the selected state:

- **Last Output state**

After a power outage, the NETIO device sets each power output to the last stored state of this one output. The current state of each power output (socket/power outlet) is internally stored every 8 seconds.

Note: **Function Scheduler** is checked in Power-Up initialization. When enabled, it can affect one or more power output stated based on current time and date.

Custom based **Lua scripts** can affect output stated too.

Energy metering variables

Since NETIO fw 3.0.0 and later, there are 23 variables available for NETIO energy metering.

Parameters for each power output:

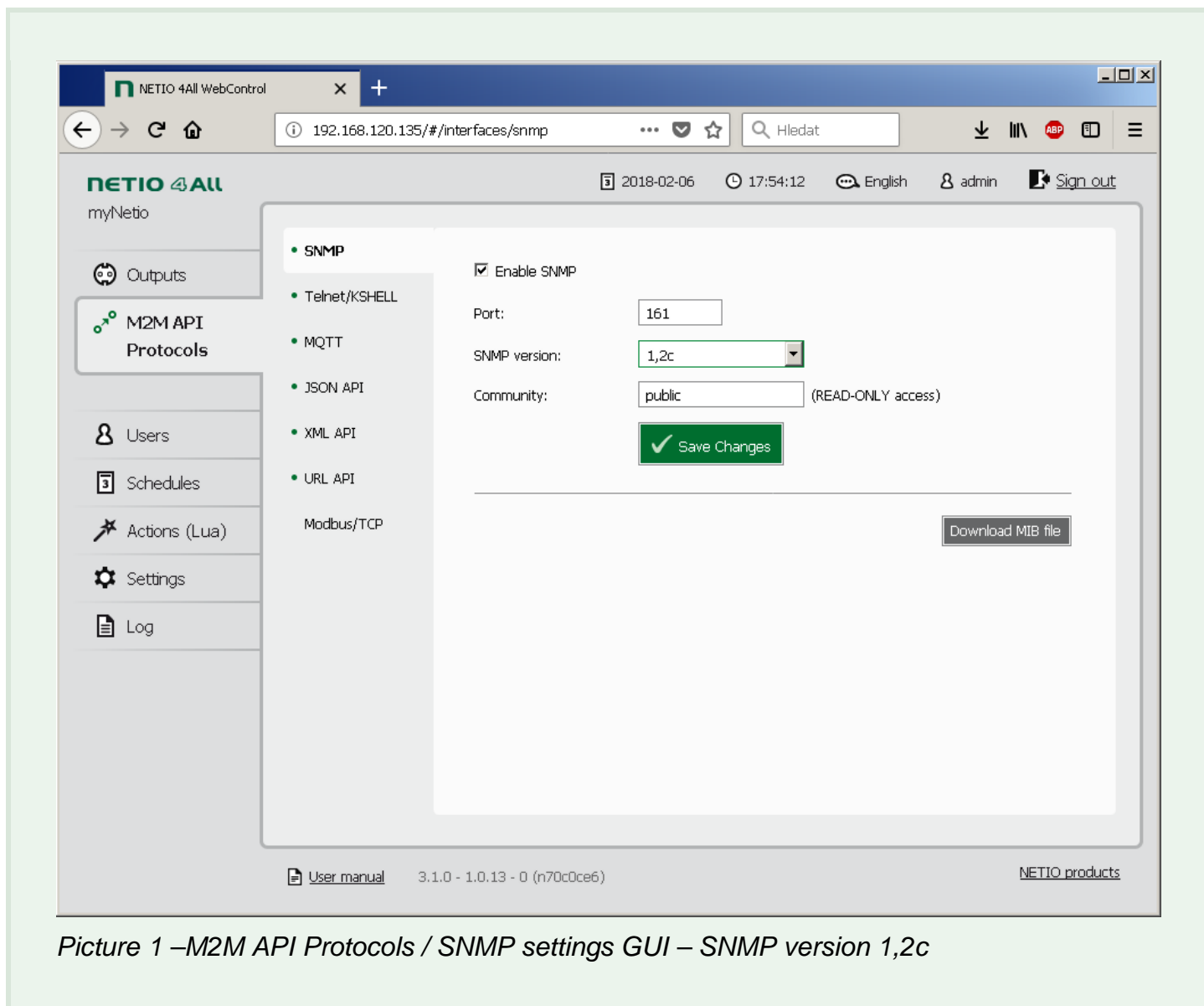
Variable	Unit	Description
4x Current	mA	Instantaneous current for the specific power output
4x PowerFactor	-	Instantaneous Power Factor for the specific power output
4x Load	W	Instantaneous load for the specific power output
4x Energy	Wh	Instantaneous Energy counter value for the specific power output

Parameters for the whole NETIO device:

Variable	Unit	Description
1x Voltage	V	Instantaneous voltage
1x Frequency	Hz	Instantaneous frequency
1x TotalCurrent	mA	Instantaneous total current through all power outputs
1x OverallPowerFactor	-	Instantaneous Power Factor – weighted average from all meters
1x TotalLoad	W	Total Load of all power outputs (device's own internal consumption is not included)
1x TotalEnergy	Wh	Instantaneous value of the Total Energy counter
1x EnergyStart	-	Date and time of the last reset of all energy counters

NETIO WEB configuration

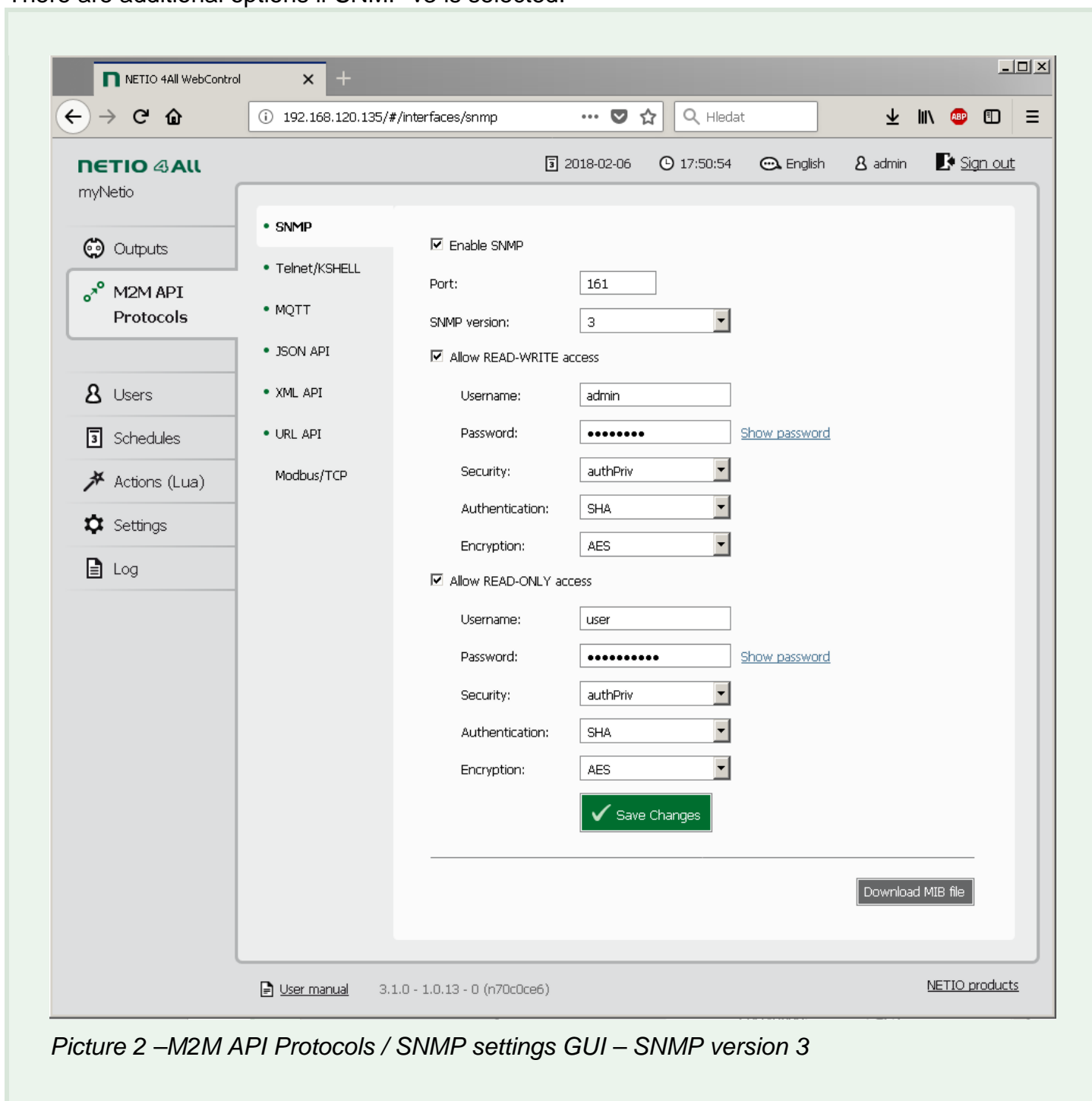
M2M API protocols can be enabled and configured only over the web administration – select “M2M API Protocols” in the left-hand side menu and then select the “SNMP” tab.



Picture 1 –M2M API Protocols / SNMP settings GUI – SNMP version 1,2c

- **Enable SNMP** – Enable/disable M2M API protocol
- **Port** – socket where SNMP is responding
- **SNMP version**
 - **1,2c** – SNMP v1 / v2c, allow only monitoring
 - **3** – SNMP v3, using authorization and encryption and allow monitoring and control
- **Download MIB file (button)** – download of NETIO MIB file

There are additional options if SNMP v3 is selected:



Picture 2 –M2M API Protocols / SNMP settings GUI – SNMP version 3

- **Allow READ-WRITE access** – enable/disable user with rights for monitoring and control
 - **Username** – user credential for user with read/write access
 - **Password** – password credential for user with read/write access
 - **Security** – USM (User-based Security Model) for read/write access
 - **Authentication** – authentication protocols for read/write access [SHA / MD5]
 - **Encryption** – privacy protocols for read/write access [AES / DES]

- **Allow READ-ONLY access** – enable/disable user with rights for only monitoring
 - **Username** – user credential for user with read-only access
 - **Password** – password credential for user with read-only access
 - **Security** – USM (User-based Security Model) for read-only access
 - **Authentication** – authentication protocols for read-only access [SHA / MD5]
 - **Encryption** – privacy protocols for read-only access [AES / DES]

Note: NETIO SNMP M2M Protocol is using “Password” as “authentication protocol pass phrase” and “privacy protocol pass phrase”.

NETIO SNMP protocol structure

NETIO MIB is available for download at the web administration – “M2M API Protocols” / “SNMP” tab.

Following standard MIBs are also required:

- SNMPv2-SMI
- SNMPv2-TC

Monitoring (read)

Object OID x – output number (1, 2, 3, 4)	Type	Value example	Note
netioOutputID.x 1.3.6.1.4.1.47952.1.1.1.1.x	INTEGER	1	
netioOutputName.x 1.3.6.1.4.1.47952.1.1.1.2.x	STRING	output_1	Based on user defined name
netioOutputState.x 1.3.6.1.4.1.47952.1.1.1.3.x	INTEGER	off(0), on(1)	
netioOutputStateString.x 1.3.6.1.4.1.47952.1.1.1.4.x	STRING	"off", "on"	
netioOutputLoad.x 1.3.6.1.4.1.47952.1.1.1.25.x	INTEGER	24	[W]
netioOutputEnergy.x 1.3.6.1.4.1.47952.1.1.1.26.x	INTEGER	13	[Wh]
netioOutputEnergyStart.x 1.3.6.1.4.1.47952.1.1.1.27.x	DateAndTime	2017-6-23,5:47:3.0,+0:0	Initial date and time. UTC based *1
netioOutputCurrent.x 1.3.6.1.4.1.47952.1.1.1.28.x	INTEGER	195	[mA]
netioOutputPowerFactor.x 1.3.6.1.4.1.47952.1.1.1.29.x	INTEGER	534	Current power factor * 1000

netioVoltage 1.3.6.1.4.1.47952.1.2.1	INTEGER	239100	Voltage in the power grid [mV]
netioFrequency 1.3.6.1.4.1.47952.1.2.2	INTEGER	49900	Frequency in the power grid [mHz]
netioTotalCurrent 1.3.6.1.4.1.47952.1.2.3	INTEGER	195	[mA]
netioOverallPowerFactor 1.3.6.1.4.1.47952.1.2.4	INTEGER	534	Current power factor * 1000
netioTotalLoad 1.3.6.1.4.1.47952.1.2.5	INTEGER	24	[W]
netioTotalEnergy 1.3.6.1.4.1.47952.1.2.6	INTEGER	13	[Wh]
netioEnergyStart 1.3.6.1.4.1.47952.1.2.7	DateAndTime	2017-6-23,5:47:3.0,+0:0	Initial date and time. UTC based

*1: The start time what Energy (cumulated consumption) is counted from. The value is the same for all outputs.

Note: Items/values related to metering (Voltage, Frequency, Current, PowerFactor, Load and Energy, etc.) are available only for the NETIO 4All model.

Control (write)

Object OID x – output number (1, 2, 3, 4)	Type	Value	Action
netioOutputAction.x 1.3.6.1.4.1.47952.1.1.1.5.x	INTEGER (i)	0 1 2 3 4 5	Turn OFF Turn ON Short OFF delay (restart) Short ON delay Toggle (invert the state) No change

NETIO MIB (version 1.05):

```

NETIO-PRODUCTS-NETIO-MIB DEFINITIONS ::= BEGIN

--      NETIO-MIB 1.05
--
--      History:
--      1.00   13.7.2016  Bretislav Bakala - Created
--      1.01   27.3.2017  Petr Kristal - MODULE-IDENTITY capitals change
--      1.02   27.5.2017  Jiri Zouhar - Added voltage, frequency, current and power

```


factor.

```
-- 1.03 8.12.2017 Jiri Zouhar - Fixed outlet power factor range & naming.
netioGlobalMeasure and netioOutletTable format definition fixed.
-- 1.04 8.12.2017 Jiri Zouhar - Synchronized naming conventions with other
protocols
-- 1.05 6.2.2018 Jiri Zouhar - Change outlet to output in all names
--
-- This is an MIB file for Netio product family - programable and remote
controlable LAN/WiFi power sockets
-- www.netio-products.com
```

IMPORTS

```
MODULE-IDENTITY, enterprises, OBJECT-TYPE FROM SNMPv2-SMI
DateAndTime FROM SNMPv2-TC
```

;

netioProducts MODULE-IDENTITY

```
LAST-UPDATED "201703270000Z"
ORGANIZATION "www.netio-products.com"
CONTACT-INFO
    "postal: NETIO products a.s.
        Mezi vodami 1955/19
        CZ 14300, Praha 4, Modrany

        email: info@netio.eu"
DESCRIPTION "Netio specific data structures"
REVISION "201703270000Z"
DESCRIPTION "NETIO products module"
 ::= { enterprises 47952 }
```

--

-- top level structure

--

```
netio4 OBJECT IDENTIFIER ::= { netioProducts 1 }
```

netioOutputTable OBJECT-TYPE

```
SYNTAX SEQUENCE OF NetioOutputEntry
MAX-ACCESS not-accessible
STATUS current
```

DESCRIPTION

"This table contains current status of Netio outputs."

::= { netio4 1 }

netioOutputEntry OBJECT-TYPE

SYNTAX NetioOutputEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A row describing one netio output."

INDEX { netioOutputID }

::= {netioOutputTable 1 }

NetioOutputEntry ::= SEQUENCE {

netioOutputID INTEGER(1..4),

netioOutputName OCTET STRING,

netioOutputState INTEGER(0..1),

netioOutputStateString OCTET STRING,

netioOutputAction INTEGER(0..5),

netioOutputLoad INTEGER(0..2147483647), -- Integer32

netioOutputEnergy INTEGER (0..9223372036854775807), -- Integer64

netioOutputEnergyStart DateAndTime,

netioOutputCurrent INTEGER(0..2147483647),

netioOutputPowerFactor INTEGER(0..1000)

}

netioOutputID OBJECT-TYPE

SYNTAX INTEGER(1..4)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Output number."

::= { netioOutputEntry 1 }

netioOutputName OBJECT-TYPE

SYNTAX OCTET STRING (SIZE(0..100))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Description of the output given by user."

```
::= { netioOutputEntry 2 }
```

```
netioOutputState OBJECT-TYPE
```

```
SYNTAX      INTEGER {  
    off(0),  
    on(1)  
}
```

```
MAX-ACCESS  read-only
```

```
STATUS      current
```

```
DESCRIPTION
```

```
"Current output state."
```

```
::= { netioOutputEntry 3 }
```

```
netioOutputStateString OBJECT-TYPE
```

```
SYNTAX      OCTET STRING (SIZE(0..20))
```

```
MAX-ACCESS  read-only
```

```
STATUS      current
```

```
DESCRIPTION
```

```
"Current output state (string)."
```

```
::= { netioOutputEntry 4 }
```

```
netioOutputAction OBJECT-TYPE
```

```
SYNTAX      INTEGER {  
    off(0),  
    on(1),  
    reset(2),  
    shortOn(3),  
    switch(4),  
    idle(5)  
}
```

```
MAX-ACCESS  read-write
```

```
STATUS      current
```

```
DESCRIPTION
```

```
"Command to control output."
```

```
::= { netioOutputEntry 5 }
```

```
netioOutputLoad OBJECT-TYPE
```

```
SYNTAX      INTEGER(0..2147483647)
```

```
MAX-ACCESS  read-only
```

```
STATUS      current
```

```
DESCRIPTION
```

```
"Current power consumption in Watts. 0 when power meters are not available"  
 ::= { netioOutputEntry 25 }
```

```
netioOutputEnergy OBJECT-TYPE
```

```
SYNTAX      INTEGER(0..9223372036854775807)
```

```
MAX-ACCESS  read-only
```

```
STATUS      current
```

```
DESCRIPTION
```

```
"Cumulated consumption from netioOutputEnergyStart. In Wh. 0 when power  
meters are not available."
```

```
 ::= { netioOutputEntry 26 }
```

```
netioOutputEnergyStart OBJECT-TYPE
```

```
SYNTAX      DateAndTime
```

```
MAX-ACCESS  read-only
```

```
STATUS      current
```

```
DESCRIPTION
```

```
"Start time from which energy is counted. UTC based. for backward  
compatibility only, is the same for all outputs. use  
netioGlobalMeasure.netioEnergyStart instead."
```

```
 ::= { netioOutputEntry 27 }
```

```
netioOutputCurrent OBJECT-TYPE
```

```
SYNTAX      INTEGER(0..2147483647)
```

```
MAX-ACCESS  read-only
```

```
STATUS      current
```

```
DESCRIPTION
```

```
"Current current in mA. 0 when power meters are not available"
```

```
 ::= { netioOutputEntry 28 }
```

```
netioOutputPowerFactor OBJECT-TYPE
```

```
SYNTAX      INTEGER(0..1000)
```

```
MAX-ACCESS  read-only
```

```
STATUS      current
```

```
DESCRIPTION
```

```
"Current power factor * 1000. 0 when power meters are not available"
```

```
 ::= { netioOutputEntry 29 }
```

```
netioGlobalMeasure OBJECT IDENTIFIER ::= { netio4 2 }
```

netioVoltage OBJECT-TYPE

SYNTAX INTEGER(0..2147483647)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Voltage in the power grid in mV. 0 when power meters are not available"

::= { netioGlobalMeasure 1 }

netioFrequency OBJECT-TYPE

SYNTAX INTEGER(0..2147483647)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Frequency in the power grid in mHz. 0 when power meters are not available"

::= { netioGlobalMeasure 2 }

netioTotalCurrent OBJECT-TYPE

SYNTAX INTEGER(0..2147483647)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Total current for all outputs in mA. 0 when power meters are not available"

::= { netioGlobalMeasure 3 }

netioOverallPowerFactor OBJECT-TYPE

SYNTAX INTEGER(0..1000)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Overall power factor across all outputs. 0 when power meters are not available"

::= { netioGlobalMeasure 4 }

netioTotalLoad OBJECT-TYPE

SYNTAX INTEGER(0..2147483647)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Total load for all outputs in W. 0 when power meters are not available"

::= { netioGlobalMeasure 5 }

netioTotalEnergy OBJECT-TYPE

SYNTAX INTEGER(0..2147483647)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Total energy for all outputs in Wh starting from netioEnergyStart. 0 when power meters are not available"

::= { netioGlobalMeasure 6 }

netioEnergyStart OBJECT-TYPE

SYNTAX DateAndTime

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Start time from which energy is counted. UTC based."

::= { netioGlobalMeasure 7 }

END

NETIO 4All – snmpwalk listing

```
NETIO-PRODUCTS-NETIO-MIB::netioOutputID.1 = INTEGER: 1
NETIO-PRODUCTS-NETIO-MIB::netioOutputID.2 = INTEGER: 2
NETIO-PRODUCTS-NETIO-MIB::netioOutputID.3 = INTEGER: 3
NETIO-PRODUCTS-NETIO-MIB::netioOutputID.4 = INTEGER: 4
NETIO-PRODUCTS-NETIO-MIB::netioOutputName.1 = STRING: "output_1"
NETIO-PRODUCTS-NETIO-MIB::netioOutputName.2 = STRING: "output_2"
NETIO-PRODUCTS-NETIO-MIB::netioOutputName.3 = STRING: "output_3"
NETIO-PRODUCTS-NETIO-MIB::netioOutputName.4 = STRING: "output_4"
NETIO-PRODUCTS-NETIO-MIB::netioOutputState.1 = INTEGER: on(1)
NETIO-PRODUCTS-NETIO-MIB::netioOutputState.2 = INTEGER: on(1)
NETIO-PRODUCTS-NETIO-MIB::netioOutputState.3 = INTEGER: on(1)
NETIO-PRODUCTS-NETIO-MIB::netioOutputState.4 = INTEGER: on(1)
NETIO-PRODUCTS-NETIO-MIB::netioOutputStateString.1 = STRING: "on"
NETIO-PRODUCTS-NETIO-MIB::netioOutputStateString.2 = STRING: "on"
NETIO-PRODUCTS-NETIO-MIB::netioOutputStateString.3 = STRING: "on"
NETIO-PRODUCTS-NETIO-MIB::netioOutputStateString.4 = STRING: "on"
NETIO-PRODUCTS-NETIO-MIB::netioOutputLoad.1 = INTEGER: 0
NETIO-PRODUCTS-NETIO-MIB::netioOutputLoad.2 = INTEGER: 0
NETIO-PRODUCTS-NETIO-MIB::netioOutputLoad.3 = INTEGER: 32
NETIO-PRODUCTS-NETIO-MIB::netioOutputLoad.4 = INTEGER: 0
NETIO-PRODUCTS-NETIO-MIB::netioOutputEnergy.1 = INTEGER: 1135
NETIO-PRODUCTS-NETIO-MIB::netioOutputEnergy.2 = INTEGER: 149
NETIO-PRODUCTS-NETIO-MIB::netioOutputEnergy.3 = INTEGER: 819
NETIO-PRODUCTS-NETIO-MIB::netioOutputEnergy.4 = INTEGER: 692
NETIO-PRODUCTS-NETIO-MIB::netioOutputEnergyStart.1 = STRING: 2017-6-
23,15:47:53.0,+0:0
NETIO-PRODUCTS-NETIO-MIB::netioOutputEnergyStart.2 = STRING: 2017-6-
23,15:47:53.0,+0:0
NETIO-PRODUCTS-NETIO-MIB::netioOutputEnergyStart.3 = STRING: 2017-6-
23,15:47:53.0,+0:0
NETIO-PRODUCTS-NETIO-MIB::netioOutputEnergyStart.4 = STRING: 2017-6-
23,15:47:53.0,+0:0
NETIO-PRODUCTS-NETIO-MIB::netioOutputCurrent.1 = INTEGER: 0
NETIO-PRODUCTS-NETIO-MIB::netioOutputCurrent.2 = INTEGER: 0
NETIO-PRODUCTS-NETIO-MIB::netioOutputCurrent.3 = INTEGER: 270
NETIO-PRODUCTS-NETIO-MIB::netioOutputCurrent.4 = INTEGER: 0
NETIO-PRODUCTS-NETIO-MIB::netioOutputPowerFactor.1 = INTEGER: 0
NETIO-PRODUCTS-NETIO-MIB::netioOutputPowerFactor.2 = INTEGER: 0
NETIO-PRODUCTS-NETIO-MIB::netioOutputPowerFactor.3 = INTEGER: 500
NETIO-PRODUCTS-NETIO-MIB::netioOutputPowerFactor.4 = INTEGER: 0
```

```
NETIO-PRODUCTS-NETIO-MIB::netioVoltage = INTEGER: 234825
NETIO-PRODUCTS-NETIO-MIB::netioFrequency = INTEGER: 49900
NETIO-PRODUCTS-NETIO-MIB::netioTotalCurrent = INTEGER: 270
NETIO-PRODUCTS-NETIO-MIB::netioOverallPowerFactor = INTEGER: 500
NETIO-PRODUCTS-NETIO-MIB::netioTotalLoad = INTEGER: 32
NETIO-PRODUCTS-NETIO-MIB::netioTotalEnergy = INTEGER: 2796
NETIO-PRODUCTS-NETIO-MIB::netioEnergyStart = STRING: 2017-6-23,15:47:53.0,+0:0
End of MIB
```

NETIO 4 – snmpwalk listing

Note: In the NETIO 4 model, there are no metering values available.

```
NETIO-PRODUCTS-NETIO-MIB::netioOutputID.1 = INTEGER: 1
NETIO-PRODUCTS-NETIO-MIB::netioOutputID.2 = INTEGER: 2
NETIO-PRODUCTS-NETIO-MIB::netioOutputID.3 = INTEGER: 3
NETIO-PRODUCTS-NETIO-MIB::netioOutputID.4 = INTEGER: 4
NETIO-PRODUCTS-NETIO-MIB::netioOutputName.1 = STRING: "output_1"
NETIO-PRODUCTS-NETIO-MIB::netioOutputName.2 = STRING: "output_2"
NETIO-PRODUCTS-NETIO-MIB::netioOutputName.3 = STRING: "output_3"
NETIO-PRODUCTS-NETIO-MIB::netioOutputName.4 = STRING: "output_4"
NETIO-PRODUCTS-NETIO-MIB::netioOutputState.1 = INTEGER: off(0)
NETIO-PRODUCTS-NETIO-MIB::netioOutputState.2 = INTEGER: on(1)
NETIO-PRODUCTS-NETIO-MIB::netioOutputState.3 = INTEGER: off(0)
NETIO-PRODUCTS-NETIO-MIB::netioOutputState.4 = INTEGER: on(1)
NETIO-PRODUCTS-NETIO-MIB::netioOutputStateString.1 = STRING: "off"
NETIO-PRODUCTS-NETIO-MIB::netioOutputStateString.2 = STRING: "on"
NETIO-PRODUCTS-NETIO-MIB::netioOutputStateString.3 = STRING: "off"
NETIO-PRODUCTS-NETIO-MIB::netioOutputStateString.4 = STRING: "on"
NETIO-PRODUCTS-NETIO-MIB::netioOutputLoad.1 = INTEGER: 0
NETIO-PRODUCTS-NETIO-MIB::netioOutputLoad.2 = INTEGER: 0
NETIO-PRODUCTS-NETIO-MIB::netioOutputLoad.3 = INTEGER: 0
NETIO-PRODUCTS-NETIO-MIB::netioOutputLoad.4 = INTEGER: 0
NETIO-PRODUCTS-NETIO-MIB::netioOutputEnergy.1 = INTEGER: 0
NETIO-PRODUCTS-NETIO-MIB::netioOutputEnergy.2 = INTEGER: 0
NETIO-PRODUCTS-NETIO-MIB::netioOutputEnergy.3 = INTEGER: 0
NETIO-PRODUCTS-NETIO-MIB::netioOutputEnergy.4 = INTEGER: 0
NETIO-PRODUCTS-NETIO-MIB::netioOutputEnergyStart.1 = STRING: 1970-1-1,0:0:0.0,+0:0
NETIO-PRODUCTS-NETIO-MIB::netioOutputEnergyStart.2 = STRING: 1970-1-1,0:0:0.0,+0:0
```



```

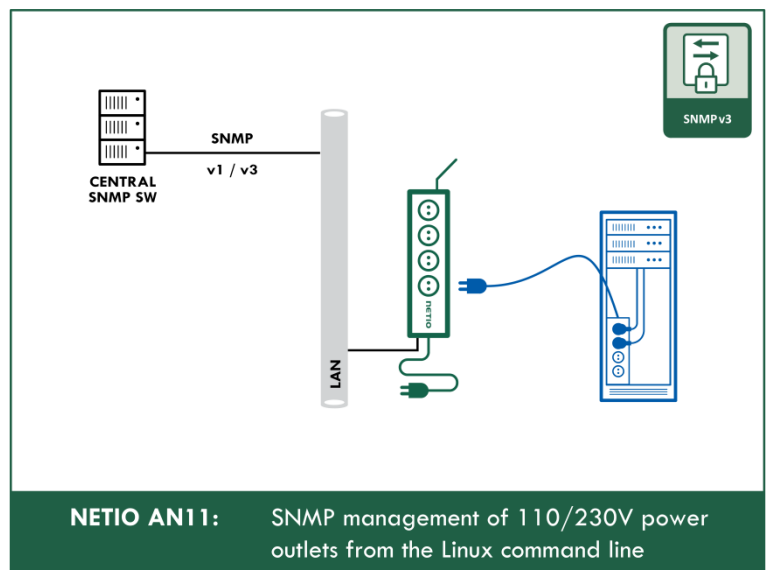
NETIO-PRODUCTS-NETIO-MIB::netioOutputEnergyStart.3 = STRING: 1970-1-1,0:0:0.0,+0:0
NETIO-PRODUCTS-NETIO-MIB::netioOutputEnergyStart.4 = STRING: 1970-1-1,0:0:0.0,+0:0
NETIO-PRODUCTS-NETIO-MIB::netioOutputCurrent.1 = INTEGER: 0
NETIO-PRODUCTS-NETIO-MIB::netioOutputCurrent.2 = INTEGER: 0
NETIO-PRODUCTS-NETIO-MIB::netioOutputCurrent.3 = INTEGER: 0
NETIO-PRODUCTS-NETIO-MIB::netioOutputCurrent.4 = INTEGER: 0
NETIO-PRODUCTS-NETIO-MIB::netioOutputPowerFactor.1 = INTEGER: 0
NETIO-PRODUCTS-NETIO-MIB::netioOutputPowerFactor.2 = INTEGER: 0
NETIO-PRODUCTS-NETIO-MIB::netioOutputPowerFactor.3 = INTEGER: 0
NETIO-PRODUCTS-NETIO-MIB::netioOutputPowerFactor.4 = INTEGER: 0
NETIO-PRODUCTS-NETIO-MIB::netioVoltage = INTEGER: 0
NETIO-PRODUCTS-NETIO-MIB::netioFrequency = INTEGER: 0
NETIO-PRODUCTS-NETIO-MIB::netioTotalCurrent = INTEGER: 0
NETIO-PRODUCTS-NETIO-MIB::netioOverallPowerFactor = INTEGER: 0
NETIO-PRODUCTS-NETIO-MIB::netioTotalLoad = INTEGER: 0
NETIO-PRODUCTS-NETIO-MIB::netioTotalEnergy = INTEGER: 0
NETIO-PRODUCTS-NETIO-MIB::netioEnergyStart = STRING: 1970-1-1,0:0:0.0,+0:0
End of MIB

```

NETIO AN (Application Note)

AN11 SNMP management of 110/230V power outlets from the command line in Windows and Linux

NETIO 4x electrical sockets (PDU) can be monitored and controlled over LAN/WiFi using SNMP. In this way, it is possible to measure the immediate consumption (NETIO 4All only) or to switch individual 110/230V sockets on or off. This AN11 Application Note shows how to perform SNMP v3 read and write operations in MS Windows or Linux.



>> Read the AN11 on www.netio-products.com

Document history

Document Revision	Publication Date	Description
1.0	11.2.2017	Initial release for FW 3.1.0, NETIO-MIB 1.05
1.1	1.12.2018	AN11 description added