

---

Version 1.0  
September 2015  
Single-Phase SubMeter: 21.61



---

# **LONMARK<sup>®</sup>**

# **Single-Phase SubMeter**

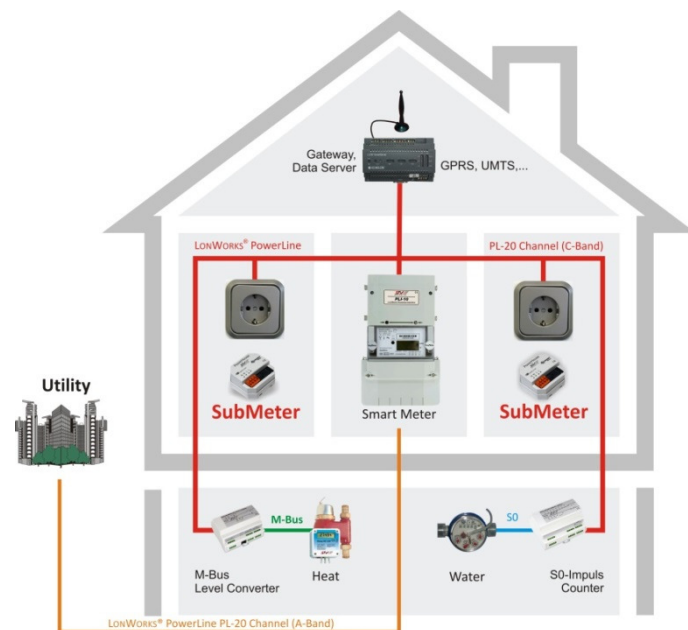
## Overview

This document describes the Functional Profile for an Electric Single-Phase SubMeter. The SubMeter is used to provide basic power metering and ON/OFF control of a load by a remote control of a relay (switch). The profile is a simplified version designed to fit in a very small memory model device.

This profile is intended for a Sub-Meter that has a fixed power circuit profile. There are no configuration settings to modify the CT, nominal voltages, or other load circuit properties. The profile focuses on providing basic electrical sub-metering.

A very important objective for this profile is to provide an interoperable device definition that makes it possible for a system integration company to deliver a system where sub meter nodes (devices) from different manufacturers can be integrated with a minimum amount of effort. In order to achieve such interoperability, the profile has defined all of the required “mandatory” variables and properties which product developers must include for certification. The expanded set of required profile elements and the reduction of optional elements have improved system interoperability and will reduce the field installation complexity for the installer.

Smart Metering and Sub Metering are fundamental terms in Home and Building Automation Systems that are driven by the requirements to have real-time information about the usage of electricity as well as water, heat or gas of many home and building automation devices.

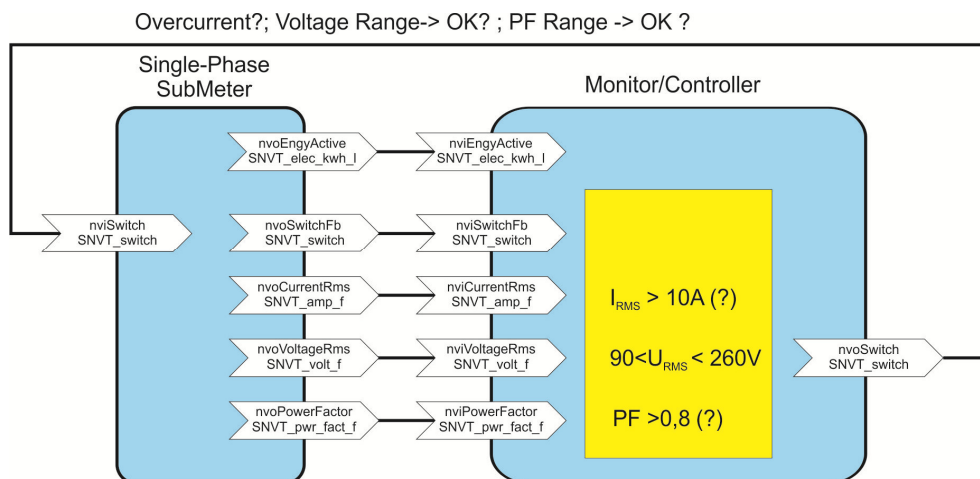


**Figure 1:** Sub Metering System - Overview

Figure 1 illustrates such a Smart Metering solution including Sub Metering. The Single-Phase SubMeter is a digital e-meter and is typically provided in a small packaged module. This small module size allows for installation in a wall box behind an outlet socket as well as on a DIN Rail. It measures the energy consumption and provides switching on/off of the ballast connected to the outlet.

## Example of Use

Figure 2 shows an example of how the Single-Phase SubMeter would be used in a system with other devices and/or functional blocks.



**Figure 2: Example SubMeter Integration Application**

The Single-Phase SubMeter Functional Block is mainly used in real-time energy measurement applications to register:

|                |   |         |
|----------------|---|---------|
| Active Energy  | – | W [kwh] |
| RMS Voltage    | – | U [V]   |
| RMS Current    | – | I [A]   |
| Power Factor   | – | PF      |
| Line Frequency | – | F [Hz]  |

Furthermore the Single-Phase SubMeter can monitor different electrical characteristic (e.g. Overcurrent, Voltage or Power Factor is out of range) and if necessary, switch on/off the connected ballasts.

---

## Authentication

For installations where the SubMeter will be used in billing systems, the system integration process should consider securing several network variables and configuration properties from unauthorized changes. The LonTalk Protocol supports a network service called authentication.

This authentication service secures network variables and all memory CP's by used of a private authentication key. Authentication service provides a layer of security for accessing and modifying network variables in the device. Of interest to lock with the authentication access is the network variable nviEngyActiveSet as it provides a means to change the totalized energy usage.

To utilize authentication, the network management tool must support configuration of authenticated service. When authentication is used, typically a single key at any one time is active in the network and is shared among all devices in the logical network. A side-effect of using authentication is the requirement to lock down (authenticate) all the network management commands which slows down the configuration process somewhat. Also, care must be taken not to lose the authentication keys as the devices that are locked may not be recoverable from a locked state.

# Functional-Block Details

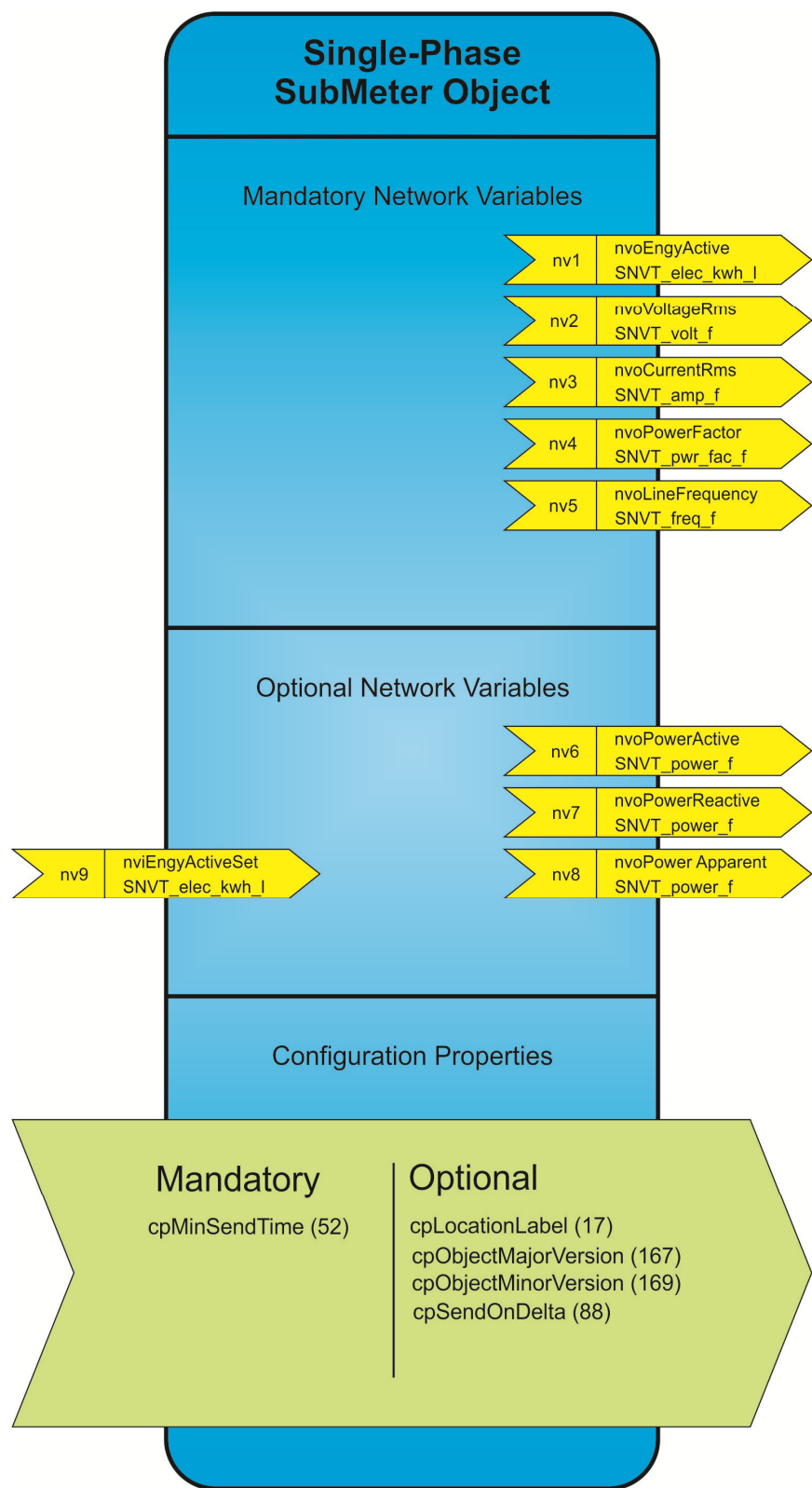
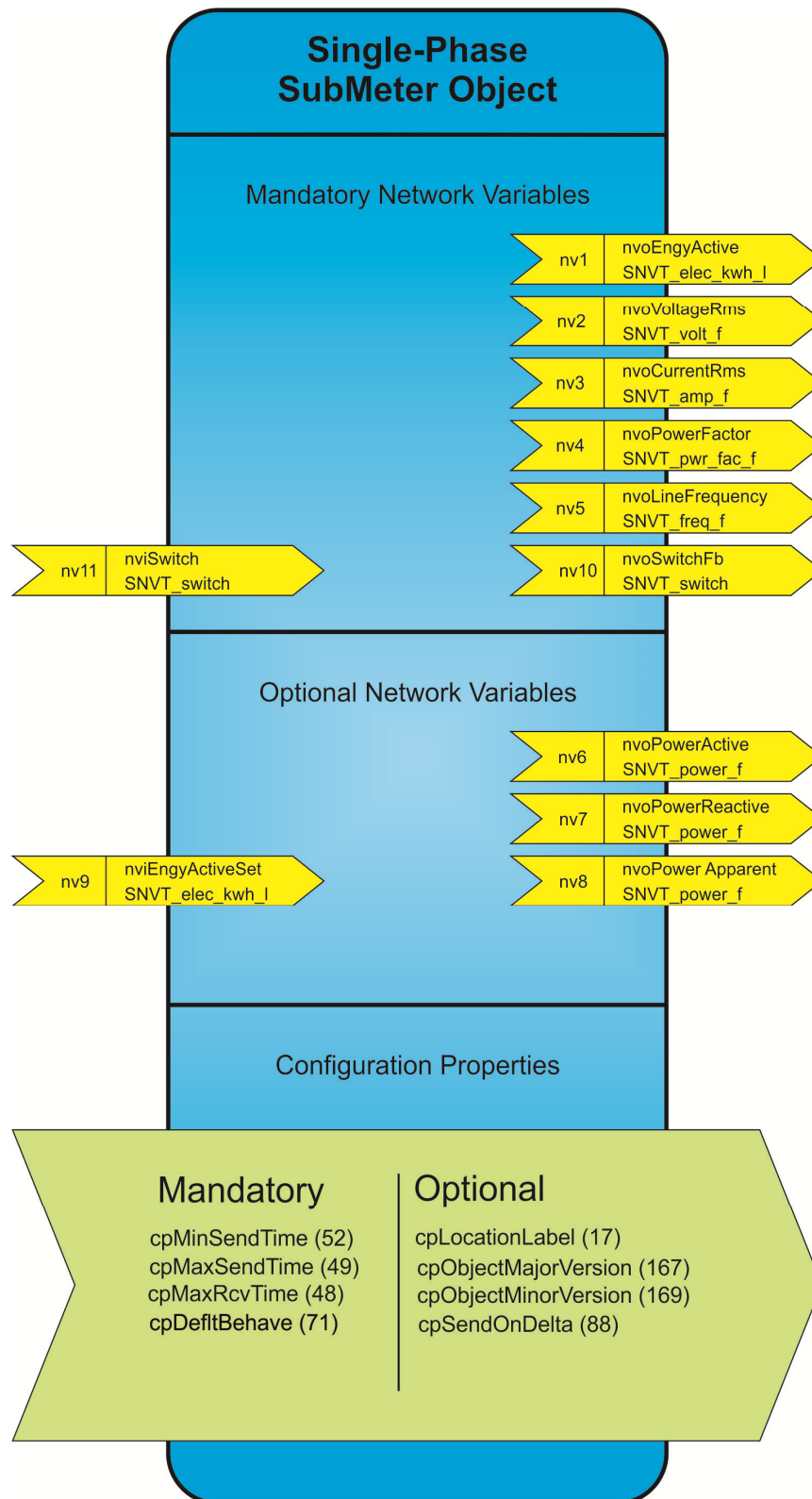


Figure 3: Standard Single-Phase SubMeter Implementation

## Functional-Block Details



**Figure 4: Single-Phase SubMeter Implementation with Switch Actuator**

**Table 1: SNVT Details**

| NV #<br>(M/O)* | Variable<br>Name | SNVT<br>Name        | SNVT<br>Index | Default<br>Value | Default<br>Service Type | Description                             |
|----------------|------------------|---------------------|---------------|------------------|-------------------------|---|
| 1 (M)          | nvoEngyActive    | SNVT_elec_kwh<br>_1 | 146           | 0.0              | UNACKD                  | Active Energy value output              |
| 2 (M)          | nvoVoltageRms    | SNVT_volt_f         | 66            | 0.0              | UNACKD                  | Voltage RMS value output                |
| 3 (M)          | nvoCurrentRms    | SNVT_amp_f          | 48            | 0.0              | UNACKD                  | Current RMS value output                |
| 4 (M)          | nvoPowerFactor   | SNVT_pwr_fact_<br>f | 99            | 0.0              | UNACKD                  | Power Factor value output               |
| 5 (M)          | nvoLineFrequency | SNVT_freq_f         | 75            | 0.0              | UNACKD                  | Line Frequency value output             |
| 6 (O)          | nvoPowerActive   | SNVT_power_f        | 57            | 0.0              | UNACKD                  | Active Power value output               |
| 7 (O)          | nvoPowerReactive | SNVT_power_f        | 57            | 0.0              | UNACKD                  | Reactive Power value output             |
| 8 (O)          | nvoPowerApparent | SNVT_power_f        | 57            | 0.0              | UNACKD                  | Apparent Power value output             |
| 9 (O)          | nviEngyActiveSet | SNVT_elec_kwh<br>_1 | 146           | 0x7FFFFFFF       | ACKD                    | Set and Reset Active Energy value input |
| 10 (M) ¥       | nvoSwitchFb      | SNVT_switch         | 95            | 100.0 1          | ACKD                    | Switch Feedback output                  |
| 11 (M) ¥       | nviSwitch        | SNVT_switch         | 95            | 100.0 1          | -                       | Switch State input                      |

Default Service Types [UNACKD = Unacknowledged Service, ACKD = Acknowledged Service, AUTH = Authentication Service ]

\* M = mandatory, O = optional

¥ Mandatory for Profile-Specific implementations requiring: “Switch Actuator Function”

(see Table 4 (page 8) - Profile-Specific Implementations)

**Table 2:** Configuration Property (CP) Details

| Associated NVs |                             |                 | SCPTminSendTime (52)<br>(Send Throttle) |                  | SCPTmaxSendTime(49)<br>(Send Heartbeat) |                  | SCPTminDeltaLevel<br>(88)<br>(SendOnDelta) |                  | SCPTmaxRcvTime(48)<br>(RcvHeartbeat) |                  | SCPTdefltBehave (71) |                  |
|----------------|-----------------------------|-----------------|---|------------------|---|------------------|--|------------------|--------------------------------------|------------------|----------------------|------------------|
|                |                             |                 | cpMinSendTimeXxxx (**                   |                  | cpMaxSendTimeXxxx (**                   |                  | cpSendOnDeltaXxxx (**                      |                  | cpMaxRcvTimeXxxx (**                 |                  | cpDbhSwitchFB        |                  |
|                |                             |                 | SNVT_time_sec (107)                     |                  | SNVT_time_sec (107)                     |                  | SNVT_lev_cont(21)                          |                  | SNVT_time_sec (107)                  |                  | SNVT_switch (95)     |                  |
| NV #<br>(M/O)* | Data Point<br>Variable Name | SNVT Name       | (M/O)*                                  | Default<br>Value | (M/O)*                                  | Default<br>Value | (M/O)*                                     | Default<br>Value | (M/O)*                               | Default<br>Value | (M/O)*               | Default<br>Value |
| 1 (M)          | nvoEngyActive               | SNVT_elec_kwh_1 | M                                       | 10               | O                                       | 300              | O  | 1(%)             | -                                    | -                | -                    | -                |
| 2 (M)          | nvoVoltageRms               | SNVT_volt_f     | M                                       | 30               | O                                       | 0.0              | O  | 5(%)             | -                                    | -                | -                    | -                |
| 3 (M)          | nvoCurrentRms               | SNVT_amp_f      | M                                       | 30               | O                                       | 0.0              | O  | 5(%)             | -                                    | -                | -                    | -                |
| 4 (M)          | nvoPowerFactor              | SNVT_pwr_fact_f | M                                       | 30               | O                                       | 0.0              | O  | 5(%)             | -                                    | -                | -                    | -                |
| 5 (M)          | nvoLineFrequency            | SNVT_freq_f     | M                                       | 30               | O                                       | 0.0              | O  | 5(%)             | -                                    | -                | -                    | -                |
| 6 (O)          | nvoPowerActive              | SNVT_power_f    | O                                       | 30               | O                                       | 0.0              | O  | 1(%)             | -                                    | -                | -                    | -                |
| 7 (O)          | nvoPowerReactive            | SNVT_power_f    | O                                       | 30               | O                                       | 0.0              | O  | 1(%)             | -                                    | -                | -                    | -                |
| 8 (O)          | nvoPowerApparent            | SNVT_power_f    | O                                       | 30               | O                                       | 0.0              | O  | 1(%)             | -                                    | -                | -                    | -                |
| 9 (O)          | nviEngyActiveSet            | SNVT_elec_kwh_1 | -                                       | -                | -                                       | -                | -  | -                | -                                    | -                | -                    | -                |
| 10 (M) ¥       | nvoSwitchFb                 | SNVT_switch     | -                                       | -                | M¥                                      | 300              | -  | -                | -                                    | -                | M¥                   | 100.0 1          |
| 11 (M) ¥       | nviSwitch                   | SNVT_switch     | -                                       | -                | -                                       | -                | -  | -                | M¥                                   | 600              | -                    | -                |

)\* M = mandatory, O = optional

¥ Mandatory for Profile-Specific implementations requiring: “Switch Actuator function”

(\*\* Xxxx = DataPoint Name → e.g.: EngyActive; VoltageRms; PowerFactor....



**Table 3:** Configuration Property (CP) - Entire Functional Block

| Man.<br>Opt. * | SCPT Name<br>NV Name<br>Type or SNVT            | SCPT<br>Index | Default<br>Value      | Associated<br>NVs **          | Description   |
|----------------|---|---------------|-----------------------|-------------------------------|---|
| Opt            | SCPTobjMajVer<br>cpObjMajVer<br>unsigned short  | 167           | 1                     | Entire<br>Functional<br>Block | Defines the major version number of the functional block  |
| Opt            | SCPTobjMinVer<br>cpObjMinVer<br>unsigned short  | 168           | 0                     | Entire<br>Functional<br>Block | Defines the minor version number of the functional block. |
| Opt            | SCPTlocation<br>cpLocation<br>SNVT_str_asc (36) | 17            | 31<br>NULLs<br>(0x00) | Entire<br>Functional<br>Block | Used to provide physical location of the device.          |

\* “Man” = mandatory, “Opt” = optional.

It should be Mandatory for CPs that are Mandatory for an NV that is also Mandatory. This is also valuable for CPs that apply to the Entire Functional Block.

\*\* List of NVs to which this configuration property applies.

An “(M)” means that the CP is Mandatory if the NV (to which it applies) is implemented. An “(O)” means that the CP is Optional if the NV (to which it applies) is implemented.

## Profile-Specific Implementations

80:00:00:XX:XX:40:00:00 (4 OR'd with other content of Usage field)

This functional profile specifies variations of itself for particular types of SubMeter Profile Implementation.

Here is the present, amendable list of SubMeter types:

| Profile-Specific<br>Usage        | Binary(b)<br>& Dec. | SPID               | Description   |
|----------------------------------|---------------------|--------------------|---|
| SubMeter                         | b000000<br>00       | :40:<br>or<br>:C0: | Standard Single-Phase<br>SubMeter                       |
| SubMeter with Switch<br>Actuator | b000001<br>01       | :41:<br>or<br>:C1: | Single-Phase SubMeter with<br>nviSwitch and nvoSwitchFb |

**Table 4:** List of SubMeter Types – Profile-Specific Implementations

---

## Mandatory Network Variables

---

### Active Energy Value Output

```
network output sd_string ("p|1") bind_info (unackd)
SNVT_elec_kwh_l nvoEngyActive;
```

This output network variable is used to send the Active Energy value of the Sub Meter.

The value of the Active Energy should be stored in a nonvolatile part of the memory. This prevent you from losing the values after a power failure. Please consider in this regard the maximum write cycles for this memory and how often you store the value.

#### *Valid Range*

The valid range of SNVT\_elec\_kwh\_l 4 byte s32\_type (signed 32-bit type),  
-2 147 483 648 to +2 147 483 647  
(Resolution 0,1 kiloWatt-hours)

#### *Default Value*

0.0

(See the Remarks "Usage of SNVT\_elec\_kwh\_l" in "LonMark® SNVT Master List" )

#### *Configuration Requirements/Restrictions*

See Table 2.

This value will be updated no faster than specified by the Send Throttle (cpMinSendTime) configuration property, if used.

#### *When Transmitted*

The output variable is transmitted:

- when polled, or
- when its value has changed and cpMinSendTime condition occurs
- when cpMaxSendTime condition occurs

#### *Default Service Type*

The default service type is unacknowledged.

---

### Voltage RMS Value Output

```
network output sd_string ("p|2") bind_info (unackd)
SNVT_volt_f nvoVoltageRms;
```

This network variable contains the Instantaneous Voltage RMS Value of the Sub Meter.

### *Valid Range*

Valid range of SNVT\_volt\_f  
4 byte floating point, (-3.4028234663853E+038 to +3.4028234663853E+038)

### *Default Value*

0.0  
See the Remarks “Usage of SNVT\_volt\_f” in “LonMark® SNVT Master List”

### *Configuration Requirements/Restrictions*

See Table 2.

This value will be updated no faster than specified by the Send Throttle (cpMinSendTime) configuration property, if used.

### *When Transmitted*

The output variable is transmitted:

- when polled, or
- when its value has changed and MinSendTime condition occurs
- when MaxSendTime condition occurs

### *Default Service Type*

The default service type is unacknowledged.

---

## **Current RMS Value Output**

```
network output sd_string ("p|3") bind_info (unackd)
SNVT_amp_f nvoCurrentRms;
```

This network variable contains the Instantaneous Current RMS Value of the Sub Meter.

### *Valid Range*

Valid range of SNVT\_amp\_f  
4 byte floating point, (-3.4028234663853E+038 to +3.4028234663853E+038)

### *Default Value*

0.0  
(See the Remarks “Usage of SNVT\_amp\_f” in “LonMark® SNVT Master List”)

## *Configuration Requirements/Restrictions*

See Table 2.

This value will be updated no faster than specified by the Send Throttle (cpMinSendTime) configuration property, if used.

### *When Transmitted*

The output variable is transmitted:

- when polled, or
- when its value has changed and MinSendTime condition occurs
- when MaxSendTime condition occurs

### *Default Service Type*

The default service type is unacknowledged.

---

## **Power Factor Value Output**

```
network output sd_string ("p|4") bind_info (unackd)
SNVT_pwr_fact_f nvoPowerFactor;
```

This network variable contains the Instantaneous Power Factor Value of the Sub Meter.

### *Valid Range*

Valid range of SNVT\_pwr\_fact\_f  
4 byte floating point, (-1 to +1)

### *Default Value*

0.0

See the Remarks “Usage of SNVT\_pwr\_f” in “LonMark® SNVT Master List”

## *Configuration Requirements/Restrictions*

See Table 2.

This value will be updated no faster than specified by the Send Throttle (cpMinSendTime) configuration property, if used.

### *When Transmitted*

The output variable is transmitted:

- when polled, or
- when its value has changed and MinSendTime condition occurs

- when MaxSendTime condition occurs

### *Default Service Type*

The default service type is unacknowledged.

---

## **Line Frequency Value Output**

```
network output sd_string ("p|5") bind_info (unackd)
SNVT_freq_f nvoLineFrequency;
```

This network variable contains the Instantaneous Line Frequency Value of the Sub Meter.

### *Valid Range*

Valid range of SNVT\_freq\_f  
4 byte floating point, (0 to +3.4028234663853E+038)

### *Default Value*

0.0  
(See the Remarks "Usage of SNVT\_freq\_f" in "LonMark® SNVT Master List")

### *Configuration Requirements/Restrictions*

See Table 2.

This value will be updated no faster than specified by the Send Throttle (cpMinSendTime) configuration property, if used.

### *When Transmitted*

The output variable is transmitted:

- when polled, or
- when its value has changed and MinSendTime condition occurs
- when MaxSendTime condition occurs

### *Default Service Type*

The default service type is unacknowledged.

---

## **Switch Feedback Value Output**

```
network output sd_string ("p|10") bind_info (ackd)
SNVT_switch nvoSwitchFb;
```

This output network variable provides the state of the switch actuator (ON or OFF)

*This NVs is **only Mandatory** for „SubMeter with Switch Actuator” Profile-Specific Usage.*

### *Valid Range*

The valid range of the state is as defined for SNVT\_switch where 0 means OFF and 1 means ON (255 - undefined - is rejected).

The 8-bit intensity value goes from 0 to 200, representing minimum to maximum intensity. (0% - 100%)

### *Default Value*

(100.0 1)

See the Remarks “Usage of SNVT\_switch” in “LonMark® SNVT Master List”

### *Configuration Requirements/Restrictions*

The transmission of this NV is regulated by the time specified in the cpMaxSendTime. Can be polled and/or propagated by an expiration of Max Send Time CP.

### *When Transmitted*

Whenever the state or intensity of the lamp actuator is requested to change, or the output variable is transmitted:

- when polled, or
- when its value has changed
- when MaxSendTime condition occurs

### *Default Service Type*

The default service type is acknowledged.

---

## **Switch Value Input**

```
network input sd_string ("p|11") SNVT_switch nviSwitch;
```

This input network variable provides a way for another device to pass switch value to the switch actuator.

*This NVs is **only Mandatory** for „SubMeter with Switch Actuator” Profile-Specific Usage.*

### *Valid Range*

The valid range of the state is as defined for SNVT\_switch where 0 means OFF and 1 means ON (255 - undefined - is rejected).

The 8-bit intensity value goes from 0 to 200, representing minimum to maximum

intensity. (0% - 100%)

### *Default Value*

(100.0 1)

See the Remarks “Usage of SNVT\_switch” in “LonMark® SNVT Master List”

---

## Optional Network Variables

---

### Active Power Value Output

```
network output sd_string ("p|6") bind_info (uackd)
SNVT_power_f nvoPowerActive;
```

This network variable contains the Instantaneous Active Power Value of the SubMeter.

#### *Valid Range*

Valid range of SNVT\_power\_f  
4 byte floating point, (-3.4028234663853E+038 to +3.4028234663853E+038)

#### *Default Value*

0.0  
(See the Remarks "Usage of SNVT\_power\_f" in "LonMark® SNVT Master List")

#### *Configuration Requirements/Restrictions*

See Table 2.

This value will be updated no faster than specified by the Send Throttle (cpMinSendTime) configuration property, if used.

#### *When Transmitted*

The output variable is transmitted:

- when polled, or
- when its value has changed and MinSendTime condition occurs
- when MaxSendTime condition occurs

#### *Default Service Type*

The default service type is unacknowledged.

---

### Reactive Power Value Output

```
network output sd_string ("p|7") bind_info (unackd)
SNVT_power_f nvoPowerReactive;
```

This network variable contains the Instantaneous Reactive Power Value of the SubMeter.



## *Valid Range*

Valid range of SNVT\_power\_f  
4 byte floating point, (-3.4028234663853E+038 to +3.4028234663853E+038)

## *Default Value*

0.0  
See the Remarks “Usage of SNVT\_power\_f” in “LonMark® SNVT Master List”

## *Configuration Requirements/Restrictions*

See Table 2.

This value will be updated no faster than specified by the Send Throttle (cpMinSendTime) configuration property, if used.

## *When Transmitted*

The output variable is transmitted:

- when polled, or
- when its value has changed and MinSendTime condition occurs
- when MaxSendTime condition occurs

## *Default Service Type*

The default service type is unacknowledged.

---

## **Apparent Power Value Output**

```
network output sd_string ("p|8") bind_info (unackd)
SNVT_power_f nvoPowerApparent;
```

This network variable contains the Instantaneous Apparent Power Value of the SubMeter.

## *Valid Range*

Valid range of SNVT\_power\_f  
4 byte floating point, (-3.4028234663853E+038 to +3.4028234663853E+038)

## *Default Value*

0.0  
See the Remarks “Usage of SNVT\_power\_f” in “LonMark® SNVT Master List”

## *Configuration Requirements/Restrictions*

See Table 2.

This value will be updated no faster than specified by the Send Throttle (cpMinSendTime) configuration property, if used.

### *When Transmitted*

The output variable is transmitted:

- when polled, or
- when its value has changed and MinSendTime condition occurs
- when MaxSendTime condition occurs

### *Default Service Type*

The default service type is unacknowledged.

---

## **Active Energy Set Input**

```
network output sd_string ("p|9") bind_info (ackd)
SNVT_elec_kwh_l nviEngyActiveSet;
```

This network variable contains the input value of the nviEngyActiveSet.

### *Valid Range*

The valid range of SNVT\_elec\_kwh\_l , 4 byte s32\_type (signed 32-bit type),  
-2 147 483 648 to +2 147 483 647  
(Resolution 0,1 kiloWatt-hours)

### *Default Value*

0x7F FF FF FF

See the Remarks "Usage of SNVT\_elec\_kwh\_l" in "LonMark® SNVT Master List"

### *Configuration Requirements/Restrictions*

See Table 2.

### *Default Service Type*

The default service type is acknowledged service.

---

## Configuration Properties

---

### Send Throttle (Mandatory)

```
config network input
sd_string("&2,i,j,k,l,m,n,o,p,0\x80,52") SNVT_time_sec
cpMinSendTime;
```

This input configuration property sets the minimum period of time that must expire before the functional block will allow updates of the following network variables to propagate across the network:

- nv1 – nvoEngyActive (Mandatory)
- nv2 – nvoVoltageRms (Mandatory)
- nv3 – nvoCurrentRms (Mandatory)
- nv4 – nvoPowerFactor (Mandatory)
- nv5 – nvoLineFrequency (Mandatory)
- nv6 – nvoPowerActive (Optional)
- nv7 – nvoPowerReactive (Optional)
- nv8 – nvoPowerApparent (Optional)

*i,j,k* are the indices of the NVs in relation to their declaration order within the device, when implemented.

#### *Valid Range*

The valid range is 0 to 6553.4 seconds.

#### *Default Value*

The default values see Table 2.

Setting SCPTminSendTime to zero disables the throttling mechanism.

#### *Configuration Requirements/Restrictions*

This CP has no modification restrictions (no restrictions). It can be modified at any time.

#### *SCPT Reference*

SCPTminSendTime (52)

---

### Receive Heartbeat (Mandatory)

```
network input config sd_string("&2,10,0\x80,48")
SNVT_time_sec cpMaxRcvTime;
```

This input configuration property sets the maximum period of time that can expire before the functional block will use the default values for the following network variables:

nv10 – nviSwitch (Mandatory – Profile-Specific Implementation “Switch Actuator”)

### *Valid Range*

The valid range is 0 to 6553.4 seconds

### *Default Value*

See Table 2.

Setting SCPTmaxRcvTime to zero disables the receive failure detect mechanism.

### *Configuration Requirements/Restrictions*

See Table 2.

## *SCPT Reference*

SCPTmaxRcvTime (48)

---

## **Send Heartbeat (Optional)**

```
network input config
sd_string("&2,i,j,k,l,m,n,o,p,q,0\x80,49")SNVT_time_sec
cpMaxSendTime;
```

This input configuration property sets the maximum period of time that can expire before the Object will automatically update the following network variables:

nv1 – nvoEngyActive (Optional)  
nv2 – nvoVoltageRms (Optional)  
nv3 – nvoCurrentRms (Optional)  
nv4 – nvoPowerFactor (Optional)  
nv5 – nvoLineFrequency (Optional)  
nv6 – nvoPowerActive (Optional)  
nv7 – nvoPowerReactive (Optional)  
nv8 – nvoPowerApparent (Optional)  
nv9 – nvoSwitchFb (Mandatory)

i.j.k...q are the indices of the NVs in relation to their declaration order within the node, when implemented.

## *Valid Range*

The valid range is 0 to 6553,4 seconds.

Setting SCPTmaxSendTime to zero disables the Heartbeat mechanism.

## *Default Value*

See Table 2.

The default value is 0, except Mandatory cpMaxSendTimeEnergyActive and cpMaxSendTimeSwitchFb - 300 Sec.

## *Configuration Requirements/Restrictions*

This CP has no modification restrictions (no\_restrictions). It can be modified at any time.

## *SCPT Reference*

SCPTmaxSendTime (49)

---

## **Send on Delta (Optional)**

```
config network input
sd_string("&2,i,j,k,l,m,n,o,p,0\x80,88") SNVT_lev_cont
cpSendOnDelta;
```

This input configuration property sets the minimum change required to force transmission of the output value to propagate across the network:

nv1 – nvoEngyActive (Optional)  
nv2 – nvoVoltageRms (Optional)  
nv3 – nvoCurrentRms (Optional)  
nv4 – nvoPowerFactor (Optional)  
nv5 – nvoLineFrequency (Optional)  
nv6 – nvoPowerActive (Optional)  
nv7 – nvoPowerReactive (Optional)  
nv8 – nvoPowerApparent (Optional)

i,j,k are the indices of the NVs in relation to their declaration order within the device, when implemented.

## *Valid Range*

The valid range for this configuration property is any value within the defined limits of the data type in question.

## *Default Value*

The default value is: see Table 2

Setting cpSndOnDelta to zero disables the sending mechanism.

## *Configuration Requirements/Restrictions*

This CP has no modification restrictions (no\_restrictions). It can be modified at any time.

## *SCPT Reference*

SCPTminDeltaLevel (88)

---

## **Location Label (Optional)**

```
network input config
sd_string("&1,p,0\x80,17")SNVT_str_asc cpLocation;
```

This configuration property can be used to provide the location of the functional block (or Sub Meter), where p is the functional-block index. The above code declaration is for providing the location of the functional block. If it is preferred, the location of the meter can be represented with the following code declaration, using the optional CP of the Node Object instead:

```
network input config sd_string("&1,0,0\x80,17")SNVT_str_asc cpLocation;
```

## *Valid Range*

Any NULL-terminated ASCII string up to 31 bytes of total length (including NULL). The string must be truncated if the length does not allow the 31st character to be the NULL (0x00).

## *Default Value*

The default value is an ASCII string containing 31 NULLs (0x00).

## *Configuration Requirements/Restrictions*

This CP has no modification restrictions (no restrictions). It can be modified at any time. Therefore this CP should not be used for identification purposes to read out the meter.

## *SCPT Reference*

SCPTlocation (17)

---

## Object Major Version (Optional)

```
network input config sd_string("&l,p,0\x84,167")unsigned  
short cpObjMajVer;
```

This configuration property can be used to provide the major version number of the functional block when implemented on a device.

### *Valid Range*

Any integer number from 0 to 255. Only 1-byte of information is accepted.

### *Default Value*

The default value is one (1).

### *Configuration Requirements/Restrictions*

This CP is a constant (const\_flg). It is not to be modified except that it is allowable to modify the value in a download of new code to the device.

### *SCPT Reference*

SCPTobjMajVer (167)

---

## Object Minor Version (Optional)

```
network input config sd_string("&l,p,0\x84,168")unsigned  
short cpObjMinVer;
```

This configuration property can be used to provide the minor version number of the functional block when implemented on a device.

### *Valid Range*

Any integer number from 0 to 255. Only 1-byte of information is accepted.

### *Default Value*

The default value is zero (0).

### *Configuration Requirements/Restrictions*

This CP is a constant (const\_flg). It is not to be modified except that it is allowable to modify the value in a download of new code to the device.

### *SCPT Reference*

SCPTobjMinVer (168)

---

## Default Behave (Optional)

```
network input config sd_string("&2,9,0\x84,71")unsigned  
short cpDbhSwitchFb;
```

This configuration property can be used to provide the default behavior of the relay within the functional block after Power On Reset when implemented on a device.

Selects which set of values will be used on power-on and communication failure, between the stated default values (0), or manufacturer-specified values (1)

### *Valid Range*

The valid range of the state is as defined for SNVT\_switch where 0 means OFF and 1 means ON (255 - undefined - is rejected).

The 8-bit intensity value goes from 0 to 200, representing minimum to maximum intensity. (0%-100%)

### *Default Value*

The default value is (100.0 1).

State of the relay after Power ON Reset = ON.

(For State >0 and Value >0 the relay is ON after Power\_ON.)

### *Configuration Requirements/Restrictions*

This CP is a constant (const\_flg). It is not to be modified except that it is allowable to modify the value in a download of new code to the device.

### *SCPT Reference*

SCPTdefltBehave (71)



---

## Key for Unresolved References

*i,j,k* are the indices of the CP-associated NVs in relation to their declaration order within the node, when implemented.

*p* is this Object's index relative to the node *sd\_string* declaration, when implemented.

---

## Data Transfer

None specified.

---

## Power-up State

There is no immediate network action on Power-up State.

---

## Boundary and Error Conditions

None specified.

---

## Additional Considerations

None specified.