Version 1.0
June 2005
Utility Meter : 22.01

LONMARK®
Functional Profile:
Utility Meter

SFPTutilityMeter

Overview

This document describes the Functional Profile of a Utility Meter functional block. The Utility Meter functional block is used to register measuring information from utility meters for remote reading; such as electricity, water, and heat meters. This is commonly known as Automated Meter Reading (AMR).

A very important objective with this profile is to provide an interoperable function that makes it possible for a utility company to have a software system where meter nodes (devices) from different manufacturers could be integrated with a minimum of effort. In order to achieve such interoperability, the profile has limited flexibility and configuration possibilities.

The Utility Meter profile supports the present meter value and a set of historical meter values; by default, the last-month meter value. The time between registering meter values in the historical register can be configured (for example, at one hour or one month).

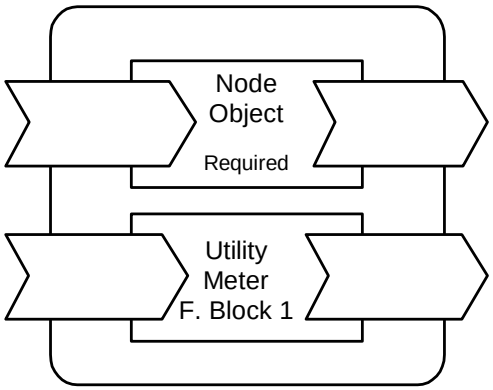

Figure 1 Device Concept

Example Usage

Here is an example of how the Utility Meter would be used in a system of other devices and/or functional blocks.

The Utility Meter functional block is mainly used in utility applications to register the cumulated value of a meter; hourly, daily, or monthly. The profile supports both integrated meter nodes, *e.g.* meters with built-in ANSI/EIA/CEA-709.1-B-2000 and EN 14908 Control Network Protocol -communication ability, and pulse-logger nodes (*i.e.* nodes that are designed to interface pulse-based meters).

All stored, historical values are actual, cumulative meter values with a time stamp and status indication.

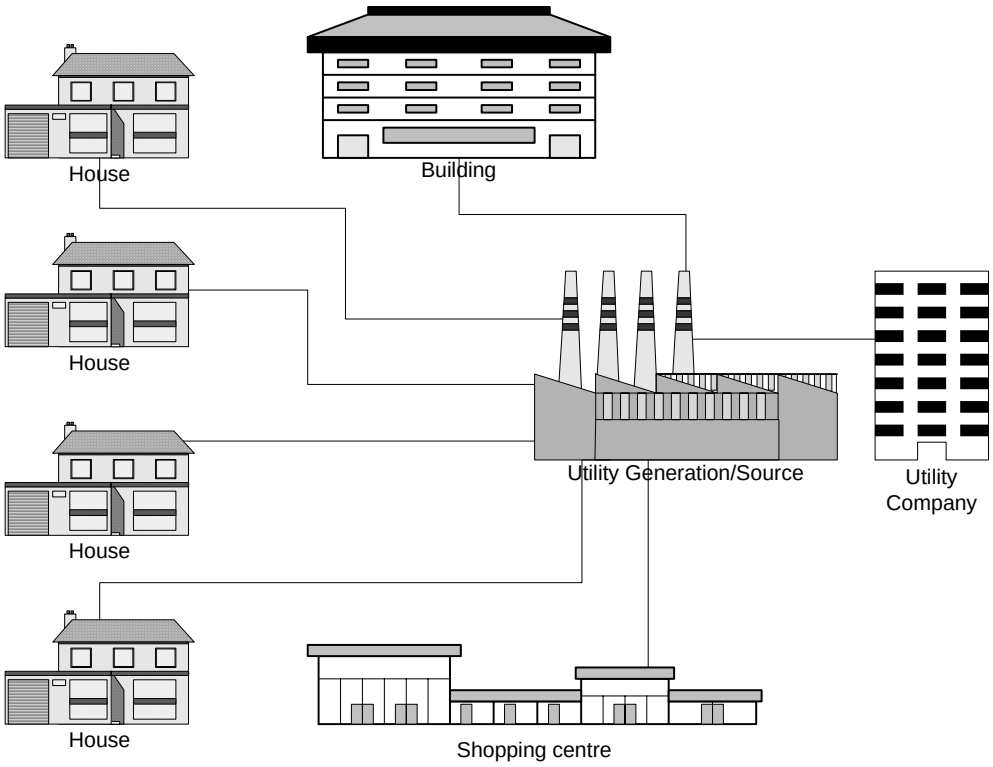

Figure 2 Example Usage of the Functional Block

Functional-Block Details

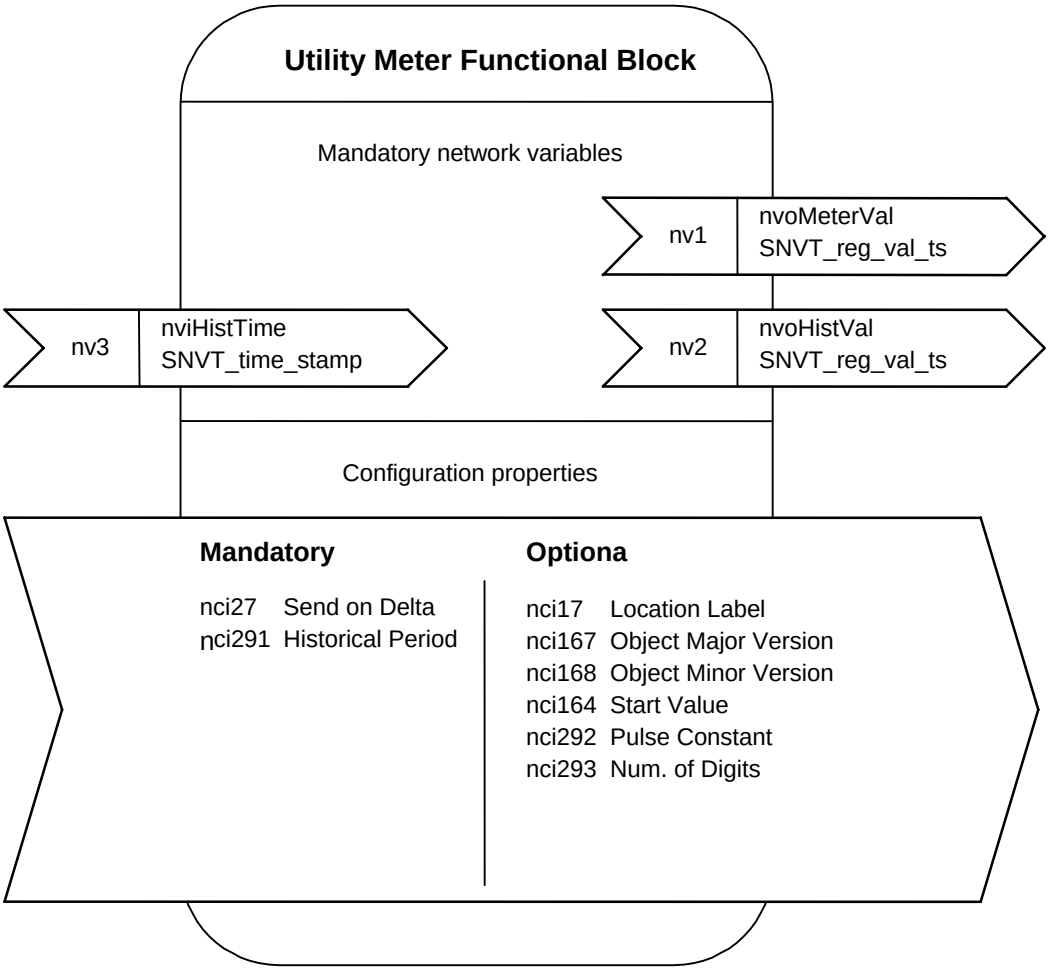

Figure 3 Functional-Block Details

Table 1 SNVT Details

NV # (M/O)*	Variable Name	SNVT Name	SNVT Index	Description
1 (M)	nvoMeterVal	SNVT_reg_val_ts	137	Meter Value Output
2 (M)	nvoHistVal	SNVT_reg_val_ts	137	Historical Value Output
3 (M)	nviHistTime	SNVT_time_stamp	84	Historical-Time Selection Input

* M = mandatory, O = optional

Table 2 SCPT Details

Man. Opt. *	SCPT Name NV Name Type or SNVT	SCPT Index	Associated NVs **	Description
Man	SCPTsndDelta nciSendOnDelta SNVT_reg_val_ts (137)	27	nv1 (M)	Maximum meter-value change before the functional block will force propagation of the NV.
Man	SCPTtimePeriod nciHistPeriod structure	291	nv2 (M)	Used to define the Historical Period of when data are saved to the meter.
Opt	SCPTlocation nciLocation SNVT_str_asc (36)	17	Entire Functional Block	Used to provide physical location of the device.
Opt	SCPTbaseValue nciStartVal SNVT_reg_val (136)	164	Entire Functional Block	Defines the Start Value for the meter register, and the number of digits to be used in the meter.
Opt	SCPTpulseValue nciPulseConst SNVT_muldiv (91)	292	Entire Functional Block	Used to scale the raw pulse value into an energy-meter value.
Opt	SCPTnumDigits nciNumDigits SNVT_count (8)	293	Entire Functional Block	Defines the number of digits of resolution, and the decimal places, for the meter.
Opt	SCPTobjMajVer nciObjMajVer unsigned short	167	Entire Functional Block	Defines the major version number of the functional block.
Opt	SCPTobjMinVer nciObjMinVer unsigned short	168	Entire Functional Block	Defines the minor version number of the functional block.

* "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.

Mandatory Network Variables

Meter Value Output

```
network output sd_string("@p|1") bind_info(ackd)
SNVT_reg_val_ts nvoMeterVal;
```

This network variable contains the present value of the meter, *i.e.*, the actual running value shown by the display on the meter. The timestamp is always given in normal time—not adjusted for daylight savings. The status information is not used (always zero).

Valid Range

Valid range of SNVT_reg_val_ts. See the Remarks "Usage of SNVT_reg_val_ts" in the *Additional Considerations* section.

Default Value

See the Remarks "Usage of SNVT_reg_val_ts" in the Additional Considerations section.

Configuration Considerations

Can be polled and/or propagated by an expiration of Send On Delta CP.

When Transmitted

The output variable is transmitted:

- when polled, or
- when the Send On Delta condition occurs.

Note that by default, the Send On Delta timer is OFF.

Default Service Type

The default service type is acknowledged.

Historical Value

```
network output sd_string("@p|2") bind_info(ackd)
SNVT_reg_val_ts nvoHistVal;
```

This network variable contains, by default, a copy of the valid meter value at the turn of the last month (*i.e.*, the value as it was at midnight between the last day in the previous month and the first day in the present month). The output timestamp of the register indicates the start of the present month. The time stamp is 00:00:00 for the time and the day is always the first of the present month. *E.g.*, 2004-11-01T00:00:00, if the present date were 2004-11-22.

This output network variable is also able to display other historical data stored by the unit. In this case, the desired output value is selected by using the nviHistTime variable. After setting nviHistTime, the nvoHistVal is updated with the data of the cumulative meter value as it was for the requested time, which then can be polled. If there does not exist a value for the requested time, the last value preceding the requested time should be sent as response.
Though the default value for historical-data storing interval is 'monthly,' the interval can be changed though the nciHistPeriod configuration property.

E.g., a request for 2004-10-30T22:**30**:00, where the value is only recorded to the register at the start of each hour, would provide on nvoHistVal the value stored for 2004-10-30T22:**00**:00. This nvoHistVal could then be polled by the requestor.

NOTE 1: It is important that the reading system (requestor) checks the timestamp of the polled value, as different data can be shown on the same variable. If there is no value earlier-in-time than the requested value, then the flag, "Illegal value request" (0x8), will be set.

NOTE 2: After sixty (60) seconds, nvoHistVal should revert back to the last turn-of-month value. This technique facilitates the collection of monthly values since it does not require the use of a command-based mechanism used to access other historical register values.

The output timestamp of the register indicates the time when the cumulative meter value was transferred to the historical register.

Valid Range

Valid range of SNVT_reg_val_ts. See the Remarks for "Usage of SNVT_reg_val_ts".

Default Value

Until the first turn-of-month, the flag, "Illegal value request" (0x8), should be set. After the first turn-of-month, the cumulative value at turn of the last month then the default value. See above and the Remarks section, "Usage of SNVT_reg_val_ts".

Configuration Considerations

Propagation of this NV is regulated by the time specified in the nciHistPeriod CP. "Loading" of a value can occur when nviHistTime is updated, but the value will not be propagated; however, it can be polled at any time.

When Transmitted

The output variable is transmitted:

- when polled, or
- when the nviHistTime input network variable is updated.
- once each month

Default Service Type

The default service type is acknowledged.

Historical Time Selection Input

```
network input sd_string("@p|3")
SNVT_time_stamp nviHistTime;
```

This input network variable controls which history value is shown on the output network variable: nvoHistVal. The value indicates a request for the meter value for a given historical time; or, if there is no value stored at the requested time, the last value immediately preceding the requested time will be transmitted.

NOTE 1: The number of available values depends on the amount of memory available in the hardware. If the time is outside of the accepted range (earlier than the first stored value), then the register output network variable, nvoHistVal, is zero and the status field indicates: "Illegal value request" (0x08) information.

Valid Range

Valid range of SNVT_time_stamp. Seconds are discarded.

Default Value

The system should provide the latest stored value that is closest to that date and time on the nvoHistVal output. For a new installed Meter the according nvoHistVal value will be invalid.

Configuration Considerations

The resulting value that is loaded into nvoHistVal is only kept for sixty (60) seconds. Therefore, the requestor (setting nviHistTime) must poll the nvoHistVal within sixty (60) seconds to obtain the information being requested.

Configuration Properties

Send on Delta (Mandatory)

```
network input config sd_string("&2,i,0\x80,27")
SNVT_reg_val_ts nciSendOnDelta;
```

This input configuration network variable defines the delta value of the meter value, nvoMeterVal. If the meter-value change exceeds the delta value, the meter value will be propagated via the network variable:

nv1 – nvoMeterVal (Mandatory)

A Mandatory CP associated with a Mandatory NV means that the CP is Mandatory for implementing this functional block (as in Table 2).

i is the index of the NV in relation to its declaration order within the device, when implemented.

Valid Range

Valid range of SNVT_reg_val_ts. The CP must use the same number of decimals, and the same unit of measure, as the variable: nvoMeterVal; otherwise, any update to this CP will be discarded.

A value of 0.0 will disable the update mechanism.

Default Value

The default value is 0.0 (no automatic update).

Configuration Requirements/Restrictions

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

SCPT Reference

SCPTsndDelta (27)

Historical Period (Mandatory)

```
network input config sd_string("&1,p,0\x80,291")
SCPTtimePeriod nciHistPeriod;
```

This input configuration network variable defines the period of time between transfer of the meter values to the historical register:

p is the index of the functional block in relation to its declaration order within the device, when implemented.

Valid Range

minutes_interval: 1, 2, 3, 5, 6, 10, 12, 15, 20, 30

hours_interval: 1, 2, 3, 4, 6, 8, 12

hour_of_day: 0 .. 23

date_of_month: 1 .. 31

Relationship examples:

nciTimePeriod.units = TP_MINUTE;
nciTimePeriod.value.minutes_interval = 10;
Meaning: Record value every ten minutes.

nciTimePeriod.units = TP_HOUR;
nciTimePeriod.value.hours_interval = 1;
Meaning: Record value every hour.

nciTimePeriod.units = TP_DAY;
nciTimePeriod.value.hour_of_day = 12;
Meaning: Record value every day at noon/midday.

nciTimePeriod.units = TP_WEEK;
nciTimePeriod.value.day_of_week = DAY_MON;
Meaning: Record value every Monday morning at midnight.

Default Value

Default data for this variable have the *nciTimePeriod.units* set to TP_MONTH and the nciTimePeriod.value.date_of_month set to one (1). This means the transfer is performed at the turn of the month at midnight of the new month.

Configuration Requirements/Restrictions

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

SCPT Reference

SCPTtimePeriod (291)

Location Label (Optional)

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

This configuration property can be used to provide the location of the functional block (or 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 nciLocation;
```

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("&1,p,0\x84,167")
unsigned short nciObjMajVer;
```

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).

The Constant flag means that all devices with the same Standard Program Identifier (SPID) will have the same value, while the Device-Specific flag attribute means that devices with an identical SPID may have different values for this configuration property.

The presence of these configuration properties within the functional block defines the major version and minor version of the functional block. The major version number must be incremented when the network interface for the functional block changes, while the minor version number must be incremented when the network interface remains the same, but the functional block has a different behavior.

SCPT Reference

SCPTobjMajVer (167)

Object Minor Version (Optional)

```
network input config sd_string("&1,p,0\ A4,168")
unsigned short nciObjMinVer;
```

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 has modification restrictions of Constant (const_flg) and Device-Specific (device_specific_flg): `\xA4`. It is not to be modified except that it *is* allowable to modify the value in a download of new code to the device.

The Constant flag means that all devices with the same Standard Program Identifier (SPID) will have the same value, while the Device-Specific flag attribute means that devices with an identical SPID may have different values for this configuration property.

The presence of these configuration properties within the functional block defines the major version and minor version of the functional block. The major version number must be incremented when the network interface for the functional block changes, while the minor version number must be incremented when the network interface remains the same, but the functional block has a different behavior.

SCPT Reference

SCPTobjMinVer (168)

Start Value (Optional)

```
network input config sd_string("&1,p,0\x81,164")
SNVT_reg_val nciStartVal;
```

This configuration property defines the start value for the meter register: nvoMeterVal. It is also used to set the number of decimal places and the unit-of-measure used by the variables: nvoMeterVal, nvoLstMonthVal and nvoHistVal.

If the nciStartVal CP is updated with a new value (not the same value it presently holds), the meter register is reset to the given value and all stored data will be lost.

NOTE 1: This optional configuration property is required for a device that is registering energy consumption by counting energy-pulses from an external meter.

p is the index of the functional block in relation to its declaration order within the device, when implemented.

Valid Range

Valid range of SNVT_reg_val.

Default Value

A value of zero. One decimal place and the units of “RVU_KWH”.

Configuration Requirements/Restrictions

This CP is to be modified only when the functional block is disabled (obj_disabl_flg). Its use can be irreparably damaging, so authentication is strongly recommended.

SCPT Reference

SCPTbaseValue (164)

Pulse and Transformer Constant (Optional)

```
network input config sd_string("&1,p,0\x81,292")
SNVT_mul_div nciPulseConst;
```

This optional configuration property is used to scale the raw pulse value to an energy-meter value. The multiplier field corresponds to a transformer constant and the divisor field corresponds to a pulse constant.

If the nciPulseConst is updated with a new value, then the next values stored (nvoLstMonthVal and nvoHistVal) should have the flag 0x1 set to indicate an unreliable value.

The meter register value is calculated by the following equation:

$$\text{MeterRegisterValue} = \frac{\text{NumberOfPulses} * \mathbf{multiplier} * 10^{\text{NumberOfDecimals}}}{\mathbf{divisor}}$$

NOTE 1: This optional configuration property is required for a device that is registering energy consumption by counting energy-pulses from an external meter.

p is the index of the functional block in relation to its declaration order within the device, when implemented.

Valid Range

Valid range of SNVT_muldiv.

Default Value

The default value is multiplier=1; divisor=1 (1/1).

Configuration Requirements/Restrictions

This CP is to be modified only when the functional block is disabled (obj_disabl_flg).

SCPT Reference

SCPTpulseValue (292)

Number of Digits on the Meter (Optional)

```
network input config sd_string("&1,p,0\x81,293")
unsigned short nciNumDigits;
```

This optional configuration property is used for setting the total number of digits on the meter. It may be used by a device connected to an external meter. The value includes the decimal digits, as specified with the nciStartVal CP. This configuration property assures that the meter register value matches the value shown on the meter display. It also specifies when the meter "rolls over" from its highest displayable value back to zero.

NOTE 1: This optional configuration property is required for a device that is registering energy consumption by counting energy-pulses from an external meter.

p is the index of the functional block in relation to its declaration order within the device, when implemented.

Valid Range

Minimum value of four (4); maximum value of nine (9).

Default Value

The default value is seven (7).

Configuration Requirements/Restrictions

This CP is to be modified only when the functional block is disabled (obj_disabl_flg).

SCPT Reference

SCPTnumDigits (293)

Key for Unresolved References

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

p is this functional block's index relative to the Device Self-Documentation String (DSDS) 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

Usage of SNVT_reg_val_ts

The output variables nvoMeterVal, and nvoHistVal are of the type SNVT_reg_val_ts. They should always show an actual meter reading value, not the consumption during a period of time. The number of decimals and unit should always be the same for these variables.

In case the node is an integrated meter the number of decimals unit should follow the actual meter value shown on the display of the meter and cannot be changed.

In case the node is a terminal unit counting S0-pulses from an external meter(s) the optional configuration properties should be used and the number of decimals and unit is set by nciStartVal, as a default one decimal and unit RVU_KWH should be used. The system reading these variables has to check for the numbers of decimals and the unit. The status bit reg_state has to be always equal to 1 (all registers are active). Status bits are needed to indicate the validity of actual data. They should be presented as masked bitmapped data. In case the node implementing this profile has been off line or has gone through a reset sequence

the flag “Internal/External error”, 0x2, has to be set in the corresponding month and historical register.

Time Stamps and Time Setting

All time stamps and time settings routines are using the normal time, i.e. not using any adjustments for day light saving.

Time setting shall be performed through the node object. This implies that it must exist a node object with a SNVT_time_stamp input network variable for time setting.

Time setting should be restricted. The rules for time setting has to follow the legislation, which can, for example, differ in between nodes for monthly and hourly stored values. The manufacturer should consider the corresponding legislation. The flag “Invalid Time”, 0x1, in SNVT_reg_val should be used. Special care has to be taken if the new time is beyond a time when the periodical value should have been stored or already has been stored.