7 Monitoring

From software version 1.2.0 the product can take own decisions and actions on values read from M-Bus slave devices and other available product information, such as GSM signal strength etc.

The product can handle up to five (5) different monitor configurations (monitor[1..5].cfg) simultaneously. The monitor configuration describes the condition that the product should evaluate, and also the commands which should be executed when the condition evaluates to true or false.

The monitor evaluates every time an M-Bus slaves is read, i.e. when running command **storevalue**. The command **monitor** can be scheduled to run down to every minute, but will not store any value as command **storevalue** does. This function is very useful when a high interval of readout of M-Bus slaves is needed for monitoring functions, but the values read should not be stored permanently.

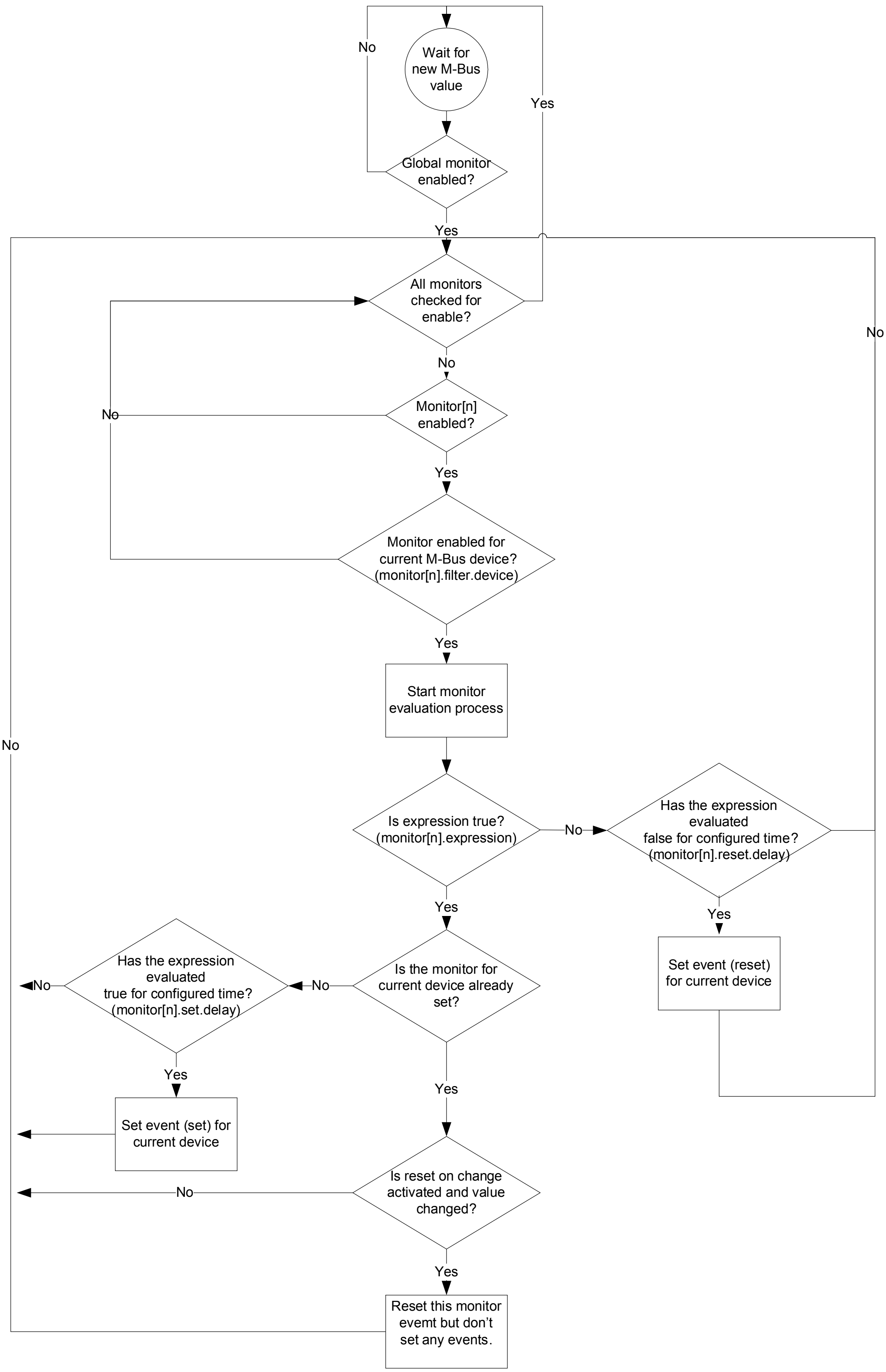
7.1 The monitoring process

The monitor process must be globally enabled by setting the configuration key common.monitor.enabled to true. When the monitor process is enabled, individual monitors can be enabled and configured. The monitors are configured the same way as normal reports and commands, using command **cfg** and command **set** to configure specific keys.

Following properties can be configured of a specific monitor:

1. The expression which will return a true or false value
2. The commands which should be executed when the expression returns true
3. The commands which should be executed when the expression returns false
4. The hysteresis (delays) before an expression triggers the event which forces the commands to run
5. The value, on change, which may reset the monitor triggered state without setting the reset event of the monitor

Please see the flowchart on next page for a logical overview of the monitoring process.


Wait for new M-Bus value
No
Yes
Global monitor enabled?
Yes
All monitors checked for enable?
No
Monitor[n] enabled?
No
Yes
Monitor enabled for current M-Bus device? (monitor[n].filter.device)
Yes
Start monitor evaluation process
No
No
Is expression true? (monitor[n].expression)
No
Has the expression evaluated false for configured time? (monitor[n].reset.delay)
Yes
Set event (reset) for current device
Yes
Has the expression evaluated true for configured time? (monitor[n].set.delay)
No
No
Is the monitor for current device already set?
Yes
Set event (set) for current device
Yes
Is reset on change activated and value changed?
No
Yes
Reset this monitor evemt but don't set any events.

7.2 Values which can be monitored

All values which are read from M-Bus slave devices with addition of all internal product variables can be used in the monitor functions. Identification of M-Bus slave device values are divided in two parts: M-Bus header values and M-Bus record values.

7.2.1 M-Bus header values

The M-Bus header values are information sent in every M-Bus telegram. Example of information in the M-Bus header is manufacturer, status information, telegram count etc. The M-Bus header values are identified using `mbus.header.<header value>`, where <header value> is the M-Bus header value which is of interest. Please see TODO for a complete listing of available M-Bus header values.

Header value	Data type	Description
cfield	Integer	M-Bus C-Field
afield	Integer	M-Bus A-Field
cifield	Integer	M-Bus CI-Field
byteordermode	Integer	M-Bus byte order mode (LSByte or MSByte)
identification	Integer	M-Bus secondary address
manufacturer	Integer	M-Bus manufacturer number
manufacturerstring	String	M-Bus three character decoded manufacturer string
version	Integer	M-Bus version/generation field
devicetypestring	Integer	M-Bus device type in decoded string
devicetype	Integer	M-Bus device type
accessnumber	Integer	M-Bus access number
status	Integer	M-Bus status code
statusstring	String	M-Bus status code decoded string
Signature	Integer	M-Bus signature

Example 1 Accessing the manufacturer string

```
mbus.header.manufacturerstring
```

7.2.2 M-Bus record values

The M-Bus record values are the DIB parts included in an M-Bus telegram, i.e. energy, flow, temperatures etc. These values are identified using `mbus.dib.<record description>`, where <record description> is the qualified name of the M-Bus record value of interest. The record description is dynamically built up during the internal M-Bus decode process, but in most cases the M-Bus slave implementation is using standard description field which can be identified within the M-Bus standard. If the description is a manufacturer specified string, the description of the M-Bus record name can be obtained by inspecting the standard installation report of the product. Please see table below for standard M-Bus record descriptions.

⚠ **IMPORTANT**

When accessing M-Bus records where description contains characters – (minus) and / (division), the character must be replaced with underscore (_).

Name
Energy
Volume
Mass
on-time
op-time
Power
volume-flow
volume-flow-ext
mass-flow
flow-temp
return-temp
diff-temp
ext-temp
Pressure
Date
Datetime
units-for-HCA
avg-duration
act-duration
fabrication-no
enhanced-id
Address
reactive-energy
cold/warm-temp-limit
cum-cnt-max-power
Credit
Debit
access-number
device-type
Manufacturer
parameter-set-id
model/version
hw-version
fw-version
other-sw-version
customer-location
Customer
access-code-user
access-code-operator
access-code-system-operator

access-code-developer
Password
error-flags-dev-spec
error-mask
digital-output
digital-input
Baudrate
response-delay-time
Retry
remote-control-dev-spec
first-storage-for-cyclic-storage
last-storage-for-cyclic-storage
size-of-storage-block
storage-interval
time-point
duration-since-last-readout
start-of-tariff
duration-of-tariff
period-of-tariff
Dimensionless
Voltage
Current
reset-counter
cum-counter
control-signal
day-of-week
week-number
time-point-of-day-change
state-of-param-activation
special-supplier-information
duration-since-last-cum
operation-time-battery
datetime-of-battery-change
day-light-saving
listening-window-management
cold/warm-temp-limit
remaining-battery-lifetime
count-meter-stop
Reserved
manufacturer-specific
no-error

too-many-DIFE
storage-number-not-impl
unit-number-not-impl
tariff-number-not-impl
function-not-impl
data-class-not-impl
data-size-not-impl
too-many-VIFE
illegal-VIF-group
illegal-VIF-exponent
VIF/DIF-mismatch
unimpl-action
no-data-available
data-overflow
data-underflow
data-error
premature-end-of-record
inc-per-input-pulse-on-input-channel-0
inc-per-input-pulse-on-input-channel-1
inc-per-output-pulse-on-input-channel-0
inc-per-output-pulse-on-input-channel-1
start-date/time-of
uncorrected-unit
acc-only-if-pos-contr
acc-of-abs-value-only-if-neg-contr
lower-limit-value
exceeds-lower-limit-value
datetime-of-begin-of-first-lower-limit-exceeded
datetime-of-end-of-first-lower-limit-exceeded
datetime-of-begin-of-last-lower-limit-exceeded
datetime-of-end-of-last-lower-limit-exceeded
upper-limit-value
exceeds-of-upper-limit-value
datetime-of-begin-of-first-upper-limit-exceeded
datetime-of-end-of-first-upper-limit-exceeded
datetime-of-begin-of-last-upper-limit-exceeded
datetime-of-end-of-last-upper-limit-exceeded
duration-of-first-lower-limit-exceeded
duration-of-last-lower-limit-exceeded
duration-of-first-upper-limit-exceeded
duration-of-last-upper-limit-exceeded

duration-of-lower-limit-exceeded
duration-of-upper-limit-exceeded
value-during-lower-limit-exceeded
leak-values
datetime-of-first-begin
datetime-of-first-end
value-during-upper-limit-exceeded
overflow-values
datetime-of-last-begin
datetime-of-last-end
future-value

Table 25 Standard description property values

7.3 Expressions

The expression (monitor[n].expression), is the actual logic which must result in a value of true or false. The expression can contain calculations (addition, subtraction, division etc) and must have an evaluation part. One of the following evaluation parts must separate the left and right side of the expression: >, >=, <, <=, ==, !=.

Example of an expression which evaluates if the power is greater than 10:

```
set monitor1.expression=$mbus.dib.power>10
```

Example of an expression which evaluates if the volume flow is less than 110:

```
set monitor1.expression=$mbus.dib.volume_flow<110
```

If the expression evaluates true for specific hysteresis (delay), the set event for this monitor and device will be set.

7.4 Set/reset delays of the monitor events

In most cases, immediate changes of a value should not result in an immediate event. This can be configured by setting the set delay and reset delay of the monitor. These settings are in seconds and be configured by the properties `monitor[n].set.delay` and `monitor[n].reset.delay`.

Example of changing the set delay of monitor1 to two minutes:

```
set monitor1.set.delay=120
```

Example of changing the reset delay to take immediate action:

```
set monitor1.reset.delay=0
```

7.5 Set commands for monitor events

When the monitor expression results in a set event or a reset event, specified commands will be executed. The commands can be a single command or a combination of commands. The command list which should be executed on a set event or a reset event can be configured by the properties `monitor[n].set.command` and `monitor[n].reset.command`. If more than one command should be executed, the commands should be separated with semicolon (;).

Example of changing the set event to send a monitor event report by e-mail:

```
set monitor1.set.command=report 1008
```

Example of changing the set event to send a monitor event report by e-mail and send breaker off to all connected CMeX40 M-Bus IO modules:

```
set monitor1.set.command=report 1008;device 254 broff
```

7.6 Using monitor templates

To enable fast and easy setup of the sophisticated monitoring functions, monitor templates can be used. The syntax of using a monitor template is the same as for setting up a report, but the monitor templates has the prefix character m.

Example of setting up monitor1 by using monitor template m1:

```
cfg monitor1 m1
```

Monitor templates are constantly developed and added and in addition customer specific templates can be developed by Elvaco to meet varying customer demands.