

Secomea Data Collection Module

■ This guide is a technical documentation of the Secomea Data Collection Module

Version: 1.0, Nov. 2018,

Applicable to SiteManagers version 9.0 or higher

Table of Contents

Change log	3
Introduction	3
Prerequisites	3
1. Configuration file	3
1.1. ConfigName	3
1.2. ConfigDescription	3
1.3. CheckpointInterval	3
1.4. AlarmsSaved	3
1.5. Collectors	3
1.5.1. CollectorName	3
1.5.2. CollectorDescription	4
1.5.3. CollectorIPAddr	4
1.5.4. CollectorPortno	4
1.5.5. Alias	4
1.5.6. Protocol	4
1.5.7. ConnectRetryMin	4
1.5.8. ConnectRetryMax	4
1.5.9. ModbusAccess	4
1.5.10. OPCUAAccess	5
1.5.11. HTTPGETAccess	6
1.5.12. Samplepoints	7
1.6. Dataservers	11
1.6.1. DatasrvName	11
1.6.2. DatasrcDescription	11
1.6.3. SampleList	11
1.6.4. IOPin	11
1.6.5. DatasrvProtocol	11
1.6.6. ConnectRetryMin	11
1.6.7. ConnectRetryMax	12
1.6.8. MQTTAWSParams	12
1.6.9. AZUREIoTHubParams	13
1.6.10. C8YParams	13
1.6.11. SemaParams	14
2. Simulator functions	15
2.1. Counter	15
2.2. Sawtooth	15
2.3. Oscillator	15
2.4. Sine	16
3. RPN expressions	16
Appendix A – Sample JSON configuration file	18
Notices	23

Change log

Version	Change log
1.0	Initial version

Introduction

The Data Collection Module (also referred to as DCM) was introduced in Secomea firmware release 9, and can be used to collect data, perform smart aggregation and then send the calculated values to a predefined cloud. For more information about this topic, please visit our website www.secomea.com/dcm

Prerequisites

- The SiteManager firmware must be version 9.0 or higher.

1. Configuration file

The configuration file for Secomea Data Collection Module is written in JSON format. Following is an explanation of the different components in the configuration file.

There is a concept of “**samplerref**”, which is a string uniquely identifying a sample. The format is **CollectorName:SampleName**.

1.1. ConfigName

Mandatory, string. Unique name, which represents the current configuration in the system.

1.2. ConfigDescription

Optional, string. A description for the configuration.

1.3. CheckpointInterval

Optional, integer. The interval (in seconds) with which all data from the internal datasample cache is saved to disk. This is useful in the event of an unstable power connection to the SiteManager. Must be higher than 60. If set to 0, this feature is disabled. If this value is not specified, it will default to 900.

1.4. AlarmsSaved

Optional, integer. The number of alarms saved in the store-and-forward database. If this value is not specified, it will default to 32.

1.5. Collectors

Mandatory, array of objects. A collector represents a device to collect data from.

1.5.1. CollectorName

Mandatory, string. Unique (within this configuration file) name of this collector. This name is used to decide which device to collect data from.

1.5.2. CollectorDescription

Optional, string. Description of this collector.

1.5.3. CollectorIPAddr

Optional, string. Provides the IP address or hostname for the device to be polled. The agent target IP address will take precedence over this.

1.5.4. CollectorPortno

Optional, integer. Provides the port number for the device to be polled on. The agent target port number will take precedence over this. If neither this or the agent target port number is defined, the appropriate port will be used, depending on the Protocol value.

1.5.5. Alias

Optional, string. This is the **CollectorName** of another collector, from which, all the parameters (expect **CollectorName**, **CollectorDescription**, **CollectorIPAddr** and **CollectorPortno**) are copied from. An alias cannot refer to another alias.

1.5.6. Protocol

Mandatory, string-enum. The protocol used to collect data. The value must be either "OPC-UA/TCP", "Modbus/TCP", "Simulator" or "Internal".

1.5.7. ConnectRetryMin

Mandatory, integer. The minimum number of seconds before trying to reestablish a connection to the device being sampled. An exponential backoff algorithm is used to retry connection establishment, starting after **ConnectRetryMin** and increasing to **ConnectRetryMax**.

1.5.8. ConnectRetryMax

Mandatory, integer. Maximum number of seconds before trying to reestablish a connection to the device being sampled.

1.5.9. ModbusAccess

Optional, object. collection of parameters needed to access a given Modbus server.

ModbusFCs

Mandatory, Array of objects. List of modbus functioncodes (FC) to be performed when collecting data if the Protocol is **Modbus/TCP**.

ModbusFCID

Mandatory, Integer. A unique ID within the collector. Used as a reference by the **Error! Reference source not found.**

ModbusFCSlaveAddress

Optional, Integer. The ModBus slave address for this access. Only used in ModBus gateway devices and should be between 0 and 247. If used in conjunction with a true ModBus device, the value should be 255. Default value is 255.

ModbusFC

Mandatory, string-enum. The type of Modbus function code used when collecting data. Must be either “coils” (FC1), “inputs” (FC2), “holding-registers” (FC3) or “input-registers” (FC4).

ModbusFCStart

Mandatory, integer. Start address on the Modbus for this action. Must be ≥ 0 .

ModbusFCCount

Mandatory, integer. Number of units (bits or registers) to read in one access. Must be ≥ 0 .

ModbusFCSampleInterval

Mandatory, integer. Number of seconds between each collection of data. Must be ≥ 0 . If 0 is used, only eventdriven polling is used.

ModbusFCTimeout

Optional, integer. Timeout in milliseconds for this Modbus request. If not present, it will follow the **ModbusTimeout** for the collector.

ModbusFCRegswap

Optional, boolean. If true, the 16bit register values retrieved will be byteswapped before being used. Default value is false.

ModbusTimeout

Optional, Integer. Timeout in milliseconds for all Modbus FC requests. Must be higher than 0. Default value is 500.

ModbusRegswap

Optional, Boolean. If true, the 16bit register values retrieved will be byteswapped before used. Default value is false.

1.5.10. OPCUAAccess

Optional, object. collection of parameters needed to access a given OPC-UA/TCP server.

OPCUASecurityPolicy

Mandatory, string-enum. The security policy for the communication channel used to collect the data from the OPC-UA server. Must be either “none”, “sign” or “signencrypt”.

OPCUACertificate

String. Name of the certificate to use for collecting data. Mandatory if OPCUASecurityPolicy is “sign” or “signencrypt”.

OPCUAUserTokenType

Mandatory, string-enum. The user authentication method used when collecting data. Must be either “anonymous”, “username”, “certificate” or “issuedtoken”.

OPCUAUsername

String. Username to be used when collecting data. Mandatory if OPCUAUserTokenType is “username”.

OPCUAPassword

String. Password to be used when collecting data. Mandatory if OPCUAUserTokenType is "username".

OPCUAUserCertificate

String. Name of user certificate to be used when collecting data. Mandatory if OPCUAUserTokenType is "certificate".

OPCUAPath

Optional, String. Additional path component to the URL used for connecting to the OPC-UA server.

OPCUATimeStampPolicy

Mandatory, string-enum. The timestamp to mark collected data with. Must be either "source", "server" or "dcm". If the polled OPC-UA server doesn't support the requested timestamp, DCM will fall back from Source > Server > DCM. This field also serves as the default value for the optional field OPCUATimeStampPolicy in sample OPCUAObject.

OPCUATimeout

Optional, integer. Timeout in milliseconds for this OPC-UA query request. Default value is 500.

1.5.11. HTTPGETAccess

Optional, object. collection of parameters needed to access a given HTTP server using GET.

HTTPGETs

Mandatory, array of objects. List of HTTP GET URLs to be performed during collection of data if the Protocol is Http/GET.

HTTPGETID

Mandatory, integer. A unique ID within this Collector for this entry. Referenced to by **Samplepoints**. Must be ≥ 0 .

HTTPGETUrl

Mandatory, string. The URL used for this access. It must adhere to the following syntax: "proto://[user[password]@]host[:port][path]". The supported "proto" values are "http", "https", "ftp" and "ftps". Note that the host, port, user and password of this URL will be replaced by **CollectorIPAddr**, **CollectorPortno**, **HTTPGETUrlUsername** and **HTTPGETUrlPassword**.

HTTPGETUrlUsername

Optional, string. Username for login to the server.

HTTPGETUrlPassword

Optional, string. Password for login to the server.

HTTPGETSampleInterval

Mandatory, integer. Number of seconds between each collection of data. Must be ≥ 0 . If 0 is used, only eventdriven polling is used.

HTTPGETUrlTimeout

Optional, integer. Timeout in milliseconds for this request. If not present, it is the value of **HTTPGETTimeout**.

HTTPGETFormat

Mandatory, string-enum. The expected type of content of the response from the server. Currently 3 values exists: "json", "xml", "raw".

HTTPGETUsername

Optional, string. Username for login to the server.

HTTPGETPassword

Optional, string. Password for login to the server.

HTTPGETUrlRootCertificate

Optional, string. Name of the ROOT certificate to use for verifying the HTTPS/FTPS server.

HTTPGETTimeout

Optional, integer. Timeout in milliseconds for all Http/Get requests. Default value is 500.

1.5.12. Samplepoints

Mandatory, array of objects. List of samples to be collected, possibly aggregated and stored in the store-and-forward database.

SampleName

Mandatory, string. Name of the sampled value.

SampleDescription

Optional, string. A description of the sampled value.

SampleUnit

Optional, string. Description of the measurement unit of the sample.

SampleGroup

Optional, string. Designates that all samplepoints in this group should be handled (displayed) together. The collection module itself places no meaning on this group affiliation.

SamplesSaved

Mandatory, integer. The number of collected values that is stored in the store-and-forward database. Value must be > 0.

OnlySaveOnChanged

Optional, boolean. If set to true, the sample is only stored if the value has changed. Default value is false.

ChangeLimit

Optional, integer or double. Describes a limit for how much changes is needed, before the value is changed. Only used if OnlySaveOnChange is true. Can be used as a simple hysteresis to filter out inaccurate (jitter) measurements.

SampleDataType

Mandatory, string-enum. The type of data collected. Possible values are: "bool" (boolean), "sbyte" (8 bit signed), "byte" (8 bit unsigned), "int16" (16 bit signed), "uint16" (16 bit unsigned), "int32" (32 bit signed), "uint32" (32 bit

unsigned), "int64" (64 bit signed), "uint64" (64 bit unsigned), "float" (32 bit), "double" (64 bit), "string" (variable text up to 24575 bytes. Includes terminating NUL), "data" (variable data up to 24575 bytes). Note that Modbus samples cannot be of type "string". Note that OPC-UA samples must have exact same type as the SampleDataType, otherwise the sample data will be ignored.

SampleLowerLimit

Optional, integer or double. Is the lower limit for the collected data. If the collected data value is less than SampleLowerLimit it is discarded.

SampleUpperLimit

Optional, integer or double. Is the upper limit for the collected data. If the collected data value is less than SampleLowerLimit it is discarded.

SampleLimitDiscard

Boolean. Controls if a sample limit violation results in the sample value being discarded. Mandatory if either SampleLowerLimit or SampleUpperLimit is present.

SampleLimitAlarm

Boolean. Controls if a sample limit violation generates an alarm. Mandatory if either SampleLowerLimit or SampleUpperLimit is present.

ModbusValue

Object. A description of how to extract a collected data value from a FC collected value. Mandatory if the protocol type is "Modbus/TCP".

FCREF

Mandatory, integer. This is a reference to the ModbusFCs element referenced by the field ModbusFCID. Value must be ≥ 0 .

FCOffset

Mandatory, integer. This is the addressing offset (relative to the referenced ModbusFC ModbusFCStart address). It must be lower than ModbusFCCount for that same ModbusFCs element.

FCDataMask

Optional, integer. The value that is bitwise AND'ed with the collected value. Must be ≥ 0 .

FCDataShift

Optional, integer. The number of bits the collected value is shifted right.

OPCUAObject

Object. A description of how to address a specific OPC-UA object, when collecting values from it. Mandatory if the protocol type is "OPC-UA/TCP".

NodeIDNameSpaceIndex

Mandatory, integer. The namespace index of the OPC-UA. Value must be between 0 and 65535.

NodeIDType

Mandatory, string-enum. The OPC-UA NodeID type. The value must be "numeric", "string", "guid" or "bytestring".

NodeIdNumeric

Integer. The OPC-UA NodeID numeric value. Only mandatory if the value of NodeIDType is "numeric".

NodeIdString

String. This is OPC-UA NodeID string value. Only mandatory if the value of NodeIDType is "string".

NodeIdGuid

String. This is OPC-UA NodeID GUID value. Only mandatory if the value of NodeIDType is "guid". The format is hex-digits with hyphens enclosed in curly brackets: {XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX} (38 characters).

NodeIdByteString

String. This is OPC-UA NodeID byte string value. Only mandatory if the value of NodeIDType is "bytestring". The format is hex-digits. Must not be an empty string and must contain an even number of hex digits.

OPCUASampleInterval

Mandatory, integer. The number of seconds between each sample of this OPC-UA object. If the interval is 0, only eventdriven polling is used.

OPCUATimeStampPolicy

Mandatory, string-enum. The timestamp to mark collected data with. Must be either "source", "server" or "dcm". If this field is not present the OPCUATimeStampPolicy filed from the OPCUAAccess object, this sample belongs to, will be used instead.

HTTPGETValue

Object. A description of how to address a specific HTTPGET value, when collecting values from it. Mandatory if the **Protocol** is "Http/Get".

HTTPGETREF

Mandatory, integer. This is a reference to the HTTPGETs element, referenced by the HTTPGETID.

Select

String. The expression which selects which element to extract the value from. Mandatory if **HTTPGETFormat** has the value "json" or "xml". When the format is "json", the string must adhere to RFC6901 JSON pointer notation. When it is "xml", it must adhere to XPath 1.0 (W3C Recommendation) notation.

InternalData

Object. A description of the internal behaviour of this sample point. Mandatory if the protocol type is "Internal".

InternalFunction

Mandatory, string-enum. This is the internal function to be performed when collecting data. Possible values are "SystemTemperature" (in centigrades Celcius), "CPULoad" (in percent), "FreeMem" (in kB), "UTCRTC" (UTC Real Time Clock broken into 8 separate values), "DigitalInput" (as 0/1), "GSMRssi" (in dBm), "GSMBer" (in percent), "GSMOperator" (text string) and "GSMTechology" (text string: "4G", "3G", "GPRS" etc.).

InternalIndex

Integer. If **InternalFunction** is "DigitalInput", this field is mandatory, and represents the index of the digital input-port to collect data from. If **InternalFunction** is "UTCRTC", this field is mandatory, and represents of the time (0 - second (0-59), 1 - minute (0-59), 2 - hour (0-23), 3 - day of the month (1-31), 4 - month (0-11), 5 - Year (-1900), 6 - day of the week (0-6, Sunday = 0), 7 - day in the year (0-365, Jan 1st = 0). Value must be ≥ 0 .

InternalSampleInterval

Mandatory, integer. The number of seconds between each sample interval. If the value is 0, only eventdriven polling will be used.

SimulatorData

Object. A description of the simulation behaviour of this sample point. Mandatory if the protocol type is "Simulator".

SimulatorFunction

Optional, string-enum. The function to be used when simulating the data. The currently possible values are: "counter", "sawtooth", "oscillator", and "sine". Default value is "counter".

SimulatorInitValue

Mandatory, double. The initial value of the simulated data.

SimulatorIncValue

Mandatory, double. The value added to the simulation data every sample interval.

SimulatorTimescale

Optional, double. The timescale (in seconds) the simulated data is being generated. The value defaults to the same value as **SimulatorSampleInterval** unless it is 0. In that case, the value will be set to 1.

SimulatorSampleInterval

Mandatory, integer. The number of seconds between each sample of this Simulator object. If set to 0, only evendriven polling is used.

Aggregation

Object. a description of how to create a new sample value by performing an aggregation function of other sample values, either physically sample values or other aggregation created sample values.

Function

Mandatory, string-enum. The aggregation function to be performed. Possible values are "max", "min", "avg", "mavg", "derived", "compute" and "eventcompute".

Values

Array of strings. A list of "sampleref" references, whose value to the aggregation function. Mandatory unless **Function** is defined as "compute" or "eventcompute".

TriggerSample

Mandatory, string. This is a sampleref to the sample that will trigger the evaluation of the aggregation function. Only when the sample referred to

is updated, will this aggregation function evaluate and create a new sample value itself.

Expression

String. An arithmetic expression using Reverse Polish Notation (RPN) with `samplerref`, numeric constants and operators separated by commas. For more details, see section XXX RPN Expressions. If **Function** is defined as “compute” or “eventcompute”, this field is mandatory.

TriggerNewInterval

Optional, integer. This is a new sample interval (in seconds) for the **TriggerSample** that should be in effect when the event is triggered. When the event is not triggered, the normal interval of the **TriggerSample** is used. Only used when the **Function** is “eventcompute”. If more than one “eventcompute” event forces a new sample interval for the same sample, the lowest new interval is used.

1.6. Dataservers

Mandatory, array of objects. A dataserver represents a cloud to send the collected data to.

1.6.1. DatasrvName

Mandatory, string. Unique name for the specific dataserver in this configuration file.

1.6.2. DatasrcDescription

Optional, string. Description of this dataserver.

1.6.3. SampleList

Optional, array of strings. This array contains the list of `samplerref`, this dataserver will deliver data for. If not specified, all sample values in this configuration will be delivered. If the **:SampleName** part of the `samplerref` is left blank, all sample values from the specified collector will be delivered.

1.6.4. IOPin

Optional, Boolean. If set to true, the Output pin configured to follow “DCM dataloss” will be controlled by this dataserver. If multiple dataservers are controlling the Output pins, then the value will be the logical OR of all dataservers data loss state. Default value is false.

1.6.5. DatasrvProtocol

Mandatory, string-enum. The protocol used to deliver the sample values to the dataservers (aka cloud servers). Possible values are “MQTT/AWS”, “AZURE/IOTHUB”, “C8Y/REST” and “SEMA/REST”.

1.6.6. ConnectRetryMin

Mandatory, integer. Minimum number of seconds before trying to establish a connection to the dataserver. An exponential backoff algorithm is used to retry connection establishment, starting after **ConnectRetryMin** and increasing to **ConnectRetryMax**.

1.6.7. ConnectRetryMax

Mandatory, integer. Maximum number of seconds before trying to reestablish a connection to the dataserver.

1.6.8. MQTTAWSPParams

Object. Describes all the parameters needed to deliver data to Amazon IOT core. mandatory if **DatasrvProtocol** is "MQTT/AWS".

BrokerAddr

Mandatory, string. The hostname or IP address of the AWS IOT core endpoint.

BrokerPort

Mandatory, integer. The TCP port to use when connecting to the dataserver. Value must be > 0 and < 65535.

RootCA

Optional, string. The name of the RootCA certificate to use for the MQTT over SSL TCP connection. If not specified, the first AWS-ROOT-CA found in the certificate repository will be used.

DeviceCert

Optional, string. The name of the device certificate to use for the MQTT over SSL TCP connection. If not specified, the first AWS Device Certificate found in the certificate repository will be used.

DevicePrivKey

Optional, string. The name of the device private key to use for the MQTT over SSL TCP connection. If not specified, the first AWS Device Private Key found in the certificate repository will be used.

PublishQos

Optional, string-enum. The Quality of Service used to publish messages to AWS. Currently "qos0" and "qos1" is supported. Default value is "qos0".

PublishInterval

Mandatory, integer. The interval in seconds between each delivery of new data from the store-and-forward database. If set to 0, the data is delivered as fast as possible.

PayloadFormat

Optional, string. The format of the payload reported over MQTT. The format is printf-like with % formats as follows:

'm' - SiteManager MAC address.

'V' - Configuration version ID.

'd' - Collector name.

's' - Sample name.

'u' - Sample unit. If undefined, an empty string.

'g' - Sample group name. If undefined, the sample name will be used.

't' - Timestamp value as an integer string.

'v' - Sample value as a string. (binary data as hex byte string)

'b' - Multiple values are delivered (in bulk) as an array, described in the **BulkElementFormat**.

'%' - A percent character. The format can be prefixed with 'j', which will perform escape sequence to data according to the RFC 7159 (JSON) standard. If not specified, the format is "{ "v" : [%jb] }".

BulkElementFormat

Optional, string. The format of the element used by the "%b" format used in **PayloadFormat**. The format is printf-like with the same as for **PayloadFormat** except "%b" is not allowed. Default format is { "ts" : %t, "%jd:%js" : %jv }.

TopicFormat

Optional, string. The format of the topic used over MQTT. The format is printf-like the same formats as **PayloadFormat**. If not specified, the format is "%m".

1.6.9. AZUREIoTHubParams

Object. Contains all the parameters needed to deliver data to the Azure IoT Hub. Mandatory if **DatasrvProtocol** is "AZURE/IOTHUB".

PublishInterval

Mandatory, integer. The interval in seconds between each delivery of new data from the store-and-forward database. If set to 0, the data is delivered as fast as possible.

ConnectName

Optional, string. The name of the connectstring to use for the connection. If not specified, the first connectstring found in the certificate repository will be used.

PayloadFormat

Optional, string. The format of the payload reported to the Azure IoT Hub. The format is the same as for the **MQTTAWSParams PayloadFormat** field.

BulkElementFormat

Optional, string. This string describes the format of the element used by the "%b" format used in the **PayloadFormat**. The format is the same as for **MQTTAWSParams PayloadFormat** field. Default value is { "ts" : %t, "%jd:%js" : %jv }.

ContentType

Optional, string. The contenttype of the values being delivered. Default value is "application/json".

ContentEncoding

Optional, string. The content encoding of the values being delivered. Default value is "utf-8".

1.6.10. C8YParams

Object. Contains all the parameters needed to deliver data to the Cumulocity Cloud. Mandatory if **DatasrvProtocol** is "C8Y/REST".

C8YRegisterURL

Mandatory, string. The URL of the Cumulocity Cloud for registering new devices.

C8YRootCAName

Mandatory, string. The name of the Cumulocity Root-CA certificate. If the certificate cannot be found, only insecure HTTP server connections can be used.

RegName

Optional, string. The name of the Cumulocity Device Registration Information to use for this connection. If not specified, the first Cumulocity Device Registration Information found in the certificate repository will be used.

PushInterval

Mandatory, integer. The interval in seconds between each delivery of new data from the store-and-forward database. If set to 0, the data is delivered as fast as possible.

1.6.11. SemaParams

Object. Contains all the parameters needed to deliver data to the Machine Advisor cloud. Mandatory if **DatasrvProtocol** is "SEMA/REST".

PushInterval

Mandatory, integer. The interval in seconds between each delivery of new data from the store-and-forward database. If set to 0, the data is delivered as fast as possible.

URLName

Optional, string. The URL to use for the Machine Advisor connection. If not specified, the first "Sema URL" found in the certificate repository will be used.

AuthName

Optional, string. The name of the Authorization string to use when connecting to the Machine Advisor dataserver. If not specified, the first Machine Advisor Authentication (header2) found in the certificate repository will be used.

PayloadFormat

Optional, string. The format of the payload reported to the Machine Advisor dataserver. The format is the same as for the **MQTTAWSParams PayloadFormat** field. Default value is "{ \"metrics\" : { \"assetName\" : \"%jd\", %jb } }". This is also referred to as "Charlie" format by Machine Advisor.

BulkElementFormat

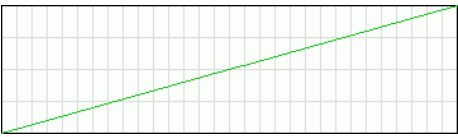
Optional, string. This string describes the format of the element used by the "%b" format used in the **PayloadFormat**. The format is the same as for **MQTTAWSParams PayloadFormat** field. Default value is \"%js\" : %jv, \"%js_timestamp\" : %E, ". This is also referred to as "Charlie" format by Machine Advisor.

2. Simulator functions

The DCM comes with a various set of simulator-functions, which are usable. For testing and demonstration purposes.

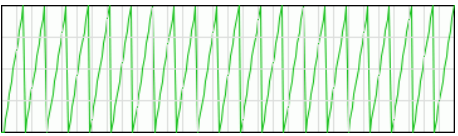
2.1. Counter

This simulator-function is a counter-function with the initial value **SimulatorInitValue** and increasing each sample-value by **SimulatorIncValue**.

Configuration	Visualized by graph
<pre>"SimulatorData": { "SimulatorFunction": "counter", "SimulatorInitValue": 1, "SimulatorIncValue": 1, "SimulatorSampleInterval": 1 }</pre>	

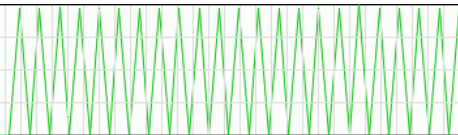
2.2. Sawtooth

This simulator-function is a counter-function with the initial value **SimulatorInitValue** and increasing each sample-value by **SimulatorIncValue**. However, when the value has increased 10 times, it will change back to **SimulatorInitValue**, and start incrementing again.

Configuration	Visualized by graph
<pre>"SimulatorData": { "SimulatorFunction": "sawtooth", "SimulatorInitValue": 1, "SimulatorIncValue": 1, "SimulatorSampleInterval": 1 }</pre>	

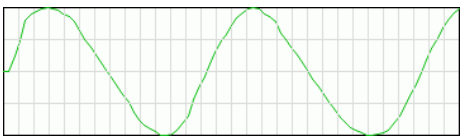
2.3. Oscillator

This simulator-function alternates between **SimulatorInitValue** and **SimulatorInitValue + SimulatorIncValue** with each sample.

Configuration	Visualized by graph
<pre>"SimulatorData": { "SimulatorFunction": "oscillator", "SimulatorInitValue": 1, "SimulatorIncValue": 1, "SimulatorSampleInterval": 1 }</pre>	

2.4. Sine

This simulator-function generates a sine curve in **SimulatorIncValue** steps over $2 \cdot \pi$ and is scaled by **SimulatorInitValue**.

Configuration	Visualized by graph
<pre>"SimulatorData": { "SimulatorFunction": "sine", "SimulatorInitValue": 1, "SimulatorIncValue": 1, "SimulatorSampleInterval": 1 }</pre>	

3. RPN expressions

The aggregation functions `compute` and `eventcompute` both have a string with an arithmetic expression, that is being evaluated, whenever the `TriggerSample` value is updated. This expression is represented in RPN (Reverse Polish Notation), which operates on a stack (max 32 values). All calculation are done in either signed 64bit integer, unsigned 64bit integer or double floating point, depending on the datatype of the aggregation sample. Once the result is stored, any overflow is ignored. The RPN is a list of operators separated by commas. The operators are:

- `samplerref` which pushes the newest value of the designated value onto the stack.
- `Constant`, which pushes the constant onto the stack.
- `"+"` (addition), which pops the top 2 values off the stack, adds them and pushes the result onto the stack.
- `"-"` (subtraction), which pops the top 2 values off the stack, subtracts them and pushes the result onto the stack.
- `"*"` (multiplication), which pops the top 2 values off the stack, multiplies them and pushes the result onto the stack.
- `"/"` (division), which pops the top 2 values off the stack, divides them and pushes the result onto the stack.
- `"%"` (modulo), which pops the top 2 values off the stack, calculates the modulo of them and pushes the result onto the stack.
- `"^"` (exponential), which pops the top 2 values off the stack, calculates the exponential of them and pushes the result onto the stack.
- `"sqrt"` (squareroot), which pops the top 1 value off the stack, calculates the squareroot of it and pushes the result onto the stack.
- `"abs"` (absolute), which pops the top 1 value off the stack, calculates the absolute value of it and pushes the result onto the stack.
- `"max"` (maximum), which pops the top 2 values off the stack, calculates the maximum of them and pushes the result onto the stack.
- `"min"` (minimum), which pops the top 2 values off the stack, calculates the minimum of them and pushes the result onto the stack.
- `"&&"` (logical and), which pops the top 2 values off the stack, calculates the logical AND of them and pushes the result onto the stack.

-
- "||" (logical or), which pops the top 2 values off the stack, calculates the logical OR of them and pushes the result onto the stack.
 - ">" (greater than), which pops the top 2 values off the stack, compares them and pushes 1 onto the stack if greater than and 0 otherwise.
 - ">=" (greater than or equal to), which pops the top 2 values off the stack, compares them and pushes 1 onto the stack if greater than or equal to and 0 otherwise.
 - "<" (less than), which pops the top 2 values off the stack, compares them and pushes 1 onto the stack if less than and 0 otherwise.
 - "<=" (less than or equal to), which pops the top 2 values off the stack, compares them and pushes 1 onto the stack if less than or equal to and 0 otherwise.
 - "==" (equal to), which pops the top 2 values off the stack, compares them and pushes 1 onto the stack if they are equal and 0 otherwise.
 - "!=" (not equal to), which pops the top 2 values off the stack, compares them and pushes 1 onto the stack if they are not equal and 0 otherwise.
 - "if" (3 values), which pops the top 3 values off the stack, pushes top-1 value if top value is non-zero, otherwise top-2 value is pushed.
 - "dropif" (1 value), which pops the top value and if it is non-zero, execution of expression will stop and no new value will be stored. NOTE: Does not push any result.

Appendix A – Sample JSON configuration file

```
{
  "ConfigName" : "Test1",
  "ConfigDescription" : "My first test configuration",
  "Collectors" : [
    {
      "CollectorName" : "Test1Collector",
      "CollectorDescription" : "Testing with ModbusTools ModbusSlave",
      "CollectorIPAddr" : "10.0.0.42",
      "CollectorPortno" : 502,
      "Protocol" : "ModBus\TCP",
      "ConnectRetryMin" : 2,
      "ConnectRetryMax" : 240,
      "ModbusAccess" : {
        "ModbusFCs" : [
          {
            "ModbusFCID" : 3,
            "ModbusFCSlaveAddress" : 1,
            "ModbusFC" : "holding-registers",
            "ModbusFCStart" : 0,
            "ModbusFCCount" : 10,
            "ModbusFCSampleInterval" : 10
          }
        ],
        "ModbusTimeout" : 1000
      },
      "SamplePoints" : [
        {
          "SampleName" : "Value1",
          "SampleDescription" : "My first register",
          "SamplesSaved" : 10,
          "SampleDataType" : "uint16",
          "ModbusValue" : {
            "FCREF" : 3,
            "FCOffset" : 0
          }
        },
        {
          "SampleName" : "Value2",
          "SampleDescription" : "My second register",
          "SamplesSaved" : 10,
          "SampleDataType" : "int32",
          "ModbusValue" : {
```

```

        "FCREF" : 3,
        "FCOffset" : 7
    },
    "SampleLowerLimit" : -8,
    "SampleUpperLimit" : 2400,
    "SampleLimitDiscard" : true,
    "SampleLimitAlarm" : true
}
]
},
{
    "CollectorName" : "Test2Collector",
    "CollectorDescription" : "Testing with Prosys OPC-UA Simulation Server",
    "Protocol" : "OPC-UA/TCP",
    "ConnectRetryMin" : 2,
    "ConnectRetryMax" : 240,
    "OPCUAAccess" : {
        "OPCUASecurityPolicy" : "none",
        "OPCUAUserTokenType" : "anonymous",
        "OPCUAPath" : "OPCUA/SimulationServer",
        "OPCUATimeStampPolicy" : "source"
    },
    "SamplePoints" : [
        {
            "SampleName" : "Value1",
            "SampleDescription" : "My first bool",
            "SamplesSaved" : 10,
            "SampleDataType" : "bool",
            "OPCUAObject" : {
                "NodeIdNameSpaceIndex" : 2,
                "NodeIdType" : "string",
                "NodeIdString" : "Value1",
                "OPCUASampleInterval" : 5
            }
        },
        {
            "SampleName" : "Value2",
            "SampleDescription" : "My first Byte",
            "SamplesSaved" : 10,
            "SampleDataType" : "byte",
            "OPCUAObject" : {
                "NodeIdNameSpaceIndex" : 2,
                "NodeIdType" : "numeric",
                "NodeIdNumeric" : 2345,

```

```

        "OPCUASampleInterval" : 2
    },
    {
        "SampleName" : "Value3",
        "SampleDescription" : "My first Int64",
        "SamplesSaved" : 10,
        "SampleDataType" : "int64",
        "OPCUAObject" : {
            "NodeIdNameSpaceIndex" : 2,
            "NodeIdType" : "guid",
            "NodeIdGuid" : "{123e4567-e89b-12d3-a456-426655440000}",
            "OPCUASampleInterval" : 27
        },
        "SampleLowerLimit" : -23,
        "SampleUpperLimit" : 24001,
        "SampleLimitDiscard" : true,
        "SampleLimitAlarm" : true
    },
    {
        "SampleName" : "Value4",
        "SampleDescription" : "My first String",
        "SamplesSaved" : 10,
        "SampleDataType" : "string",
        "OPCUAObject" : {
            "NodeIdNameSpaceIndex" : 2,
            "NodeIdType" : "string",
            "NodeIdString" : "StatusMessage",
            "OPCUASampleInterval" : 1
        },
        "SampleLimitDiscard" : false,
        "SampleLimitAlarm" : true
    },
    {
        "SampleName" : "Value5",
        "SampleDescription" : "A simulation counter",
        "SamplesSaved" : 10,
        "SampleDataType" : "int32",
        "OPCUAObject" : {
            "NodeIdNameSpaceIndex" : 5,
            "NodeIdType" : "string",
            "NodeIdString" : "Counter1",
            "OPCUASampleInterval" : 1
        }
    }
}

```

```

    },
    {
        "SampleName" : "Value3_AVG",
        "SampleDescription" : "My first Aggregation",
        "SamplesSaved" : 10,
        "SampleDataType" : "uint64",
        "Aggregation" : {
            "Function" : "avg",
            "Values" : [ "Value3" ],
            "TriggerSample" : "Value3"
        }
    },
    {
        "SampleName" : "Value3_MAX",
        "SampleDescription" : "My Second Aggregation",
        "SamplesSaved" : 10,
        "SampleDataType" : "uint64",
        "Aggregation" : {
            "Function" : "avg",
            "Values" : [ "Test1Collector:Value2", "Value2", "Value3" ],
            "TriggerSample" : "Value3"
        }
    },
    {
        "SampleName" : "Value3_Complex",
        "SampleDescription" : "My Third Aggregation",
        "SamplesSaved" : 10,
        "SampleDataType" : "uint64",
        "Aggregation" : {
            "Function" : "compute",
            "Expression" : "Test1Collector:Value2,256,*,Value3,+",
            "TriggerSample" : "Value2"
        }
    }
]
},
{
    "CollectorName" : "Another_Test1Collector",
    "CollectorDescription" : "A copy of my first fictional collector",
    "CollectorIPAddr" : "192.168.0.47",
    "CollectorPortno" : 4711,
    "Alias" : "Test1Collector"
}
],

```

```
"DataServers" : [
  {
    "DatasrvName" : "aws_iot",
    "DatasrvProtocol" : "AWS\ /MQTT",
    "ConnectRetryMin" : 2,
    "ConnectRetryMax" : 240,
    "MQTTAWSParams" : {
      "BrokerAddr" : "alcn4k05gzdtcl-ats.iot.us-east-2.amazonaws.com",
      "BrokerPort" : 8883,
      "RootCA" : "awsrootca1",
      "DeviceCert" : "awsdevcert1",
      "DevicePrivKey" : "awsdevpkey1",
      "PublishInterval" : 60,
      "TopicFormat" : "%d/%s"
    },
    "SampleList" : [ "Test1Collector:Value1", "Test2Collector:",
"Test1Collector:Value2" ]
  }
]
```

Notices

Publication and copyright

© **Copyright Secomea A/S 2018**. All rights reserved. You may download and print a copy for your own use. As a high-level administrator, you may use whatever you like from contents of this document to create your own instructions for deploying our products. Otherwise, no part of this document may be copied or reproduced in any way, without the written consent of Secomea A/S. We would appreciate getting a copy of the material you produce in order to make our own material better and – if you give us permission – to inspire other users.

Trademarks

SiteManager™, LinkManager™ and GateManager™ are trademarks of Secomea A/S. Other trademarks are the property of their respective owners.

Disclaimer

Secomea A/S reserves the right to make changes to this publication and to the products described herein without notice. The publication of this document does not represent a commitment on the part of Secomea A/S. Considerable effort has been made to ensure that this publication is free of inaccuracies and omissions but we cannot guarantee that there are none.

The following paragraph does not apply to any country or state where such provisions are inconsistent with local law:

SECOMEA A/S PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE

SECOMEA A/S SHALL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL, OR OTHER DAMAGE ALLEGED IN CONNECTION WITH THE FURNISHING OR USE OF THIS INFORMATION.