ibsysComponents

Dokumentation zum Modul ibsysComponents

Version: V2.0

20. April 2026

IBSYS GmbH
Lagerplatz 6, 8400 Winterthur
Autor: Levi Jetzer

Allgemein

Die ibsysComponents sind Funktionsblöcke, welche zur Programmierung von Niagara Steuerungen eingesetzt werden können. Hauptsächlich darin zu finden sind Bausteine für HLK-Anwendungen.

Ab Version 4.3.2.2 wurden License Credits als Lizenzeinheit eingeführt. Welche Komponente wie viele License Credits benötigt, wird im Kapitel License Credits dargestellt.

Kompatibilität

Das Modul ibsysComponents ist ab der Niagara Version N4.10 einsetzbar.

Version

Diese Dokumentation gilt ab der Modulversion 4.4.1.2.

Kontakt

Bei Fragen, Anmerkungen, Anregungen oder Fehlermeldungen kontaktieren Sie bitte unseren technischen Support:

support@ibsys.ch

www.ibsys.ch

Versionsindex

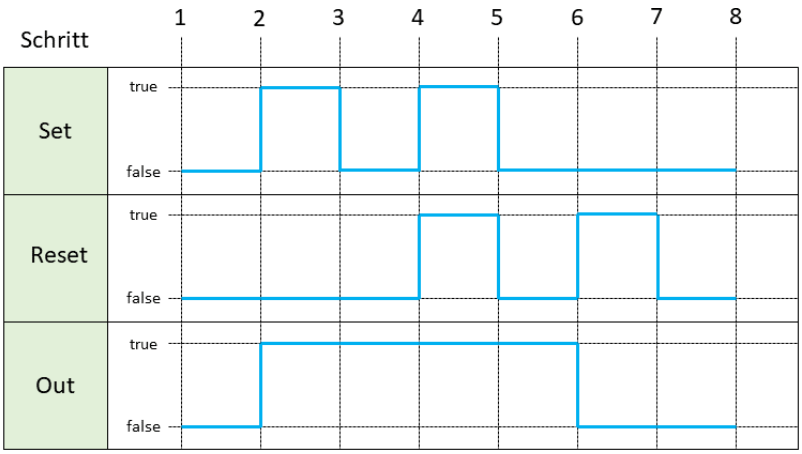
Version	Datum	Bemerkung	Autor
V1.0	31.01.2024	Erstellung	Levi Jetzer
V1.1	11.05.2024	Bearbeitet auf Modul Release V4.1.1.2	Levi Jetzer
V1.2	22.07.2024	Bearbeitet auf Modul Release V4.2.1.2; Add16, Min16, Max16, VavSetpointCalc und HistoryCsvExporter hinzugefügt	Levi Jetzer
V1.3	08.08.2024	Bearbeitet auf Modul Release V4.2.1.12, BooleanChanged, NumericChanged, EnumChanged, StringChanged	Levi Jetzer
V1.4	08.12.2025	Bearbeitet auf Modul Release V4.3.2.2, AlarmRoutingFilter & License Credits	Levi Jetzer
V2.0	20.04.2026	Bearbeitet auf Modul Release V4.4.1.2, AlarmToString, MakeSimpleJsonObject, GetJsonObject, HistoryCsvExporter, StringBase64Encoder	Levi Jetzer

Inhaltsverzeichnis

1. FlipFlops

1.1. SR

Das SR-Flipflop ist ein bistabiler Funktionsblock mit dominantem Setzen und gehört zu den standard IEC-Bausteinen. Die Funktion ist wie folgt:

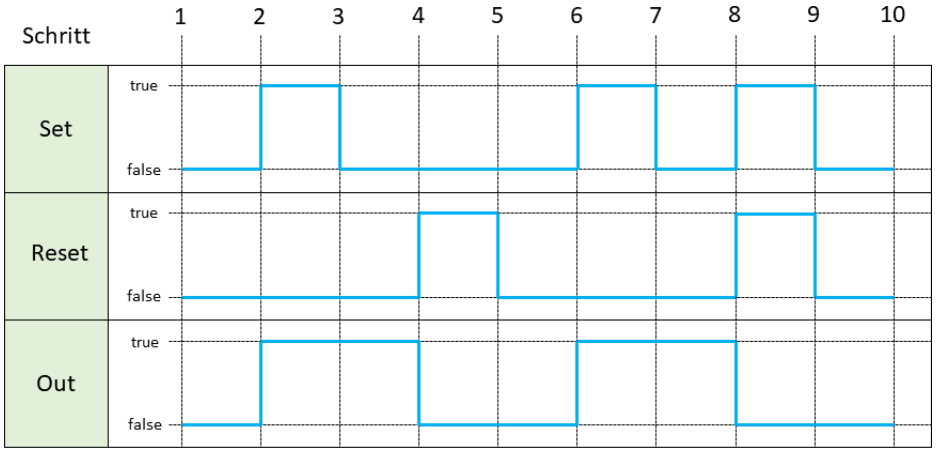


Der Wert im Out Slot wird jeweils gespeichert.

1.2. RS

Das RS-Flipflop ist ein bistabiler Funktionsblock mit dominantem Rücksetzen und gehört zu den standard IEC-Bausteinen. Die Funktion ist wie folgt:

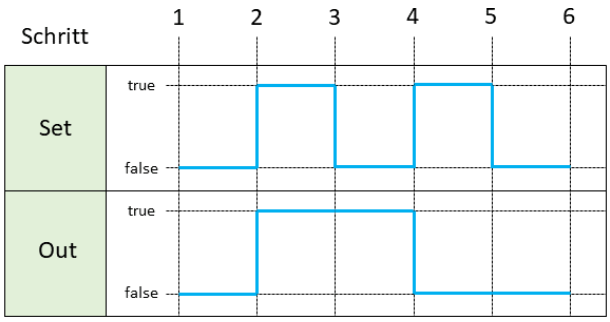
Der Wert im Out Slot wird jeweils gespeichert.



1.3. FlipFlop

Das Flipflop ist ein bistabiler Funktionsblock mit einem Eingang. Die Funktion ist wie folgt:

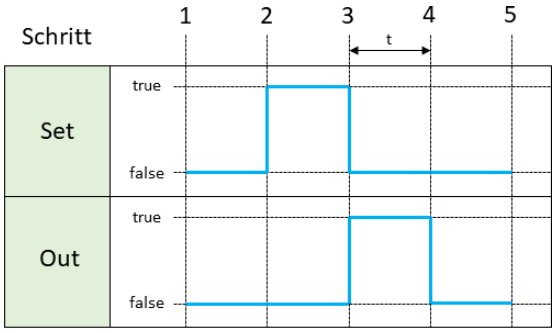
Der Wert im Out Slot wird jeweils gespeichert.



1.4. NegativeEdge

Die NegativeEdge ist eine negative Flankenerkennung. Die Funktion ist wie folgt:

Der Ausgang wir nach einer negativen Flanke während einer einstellbaren Zeit (t) auf true gesetzt.



2. Logic

2.1. And16

Der And16 ist ein gewöhnlicher AND-Baustein, verfügt jedoch über 16 Eingänge. Der Unterschied zum And aus der kitControl Palette ist weiter der Slot «Out Not», bei welchem der negierte Wert des «Out» Slot ausgegeben wird.

2.2. Or16

Der Or16 ist ein gewöhnlicher OR-Baustein, verfügt jedoch über 16 Eingänge. Der Unterschied zum Or aus der kitControl Palette ist weiter der Slot «Out Not», bei welchem der negierte Wert des «Out» Slot ausgegeben wird.

2.3. Average16

Der Average16 ist ein gewöhnlicher AVERAGE-Baustein, wie jener aus der kitControl Palette, verfügt jedoch über 16 Eingänge.

2.4. Add16

Der Add16 ist ein gewöhnlicher ADD-Baustein, wie jener aus der kitControl Palette, verfügt jedoch über 16 Eingänge.

2.5. Min16

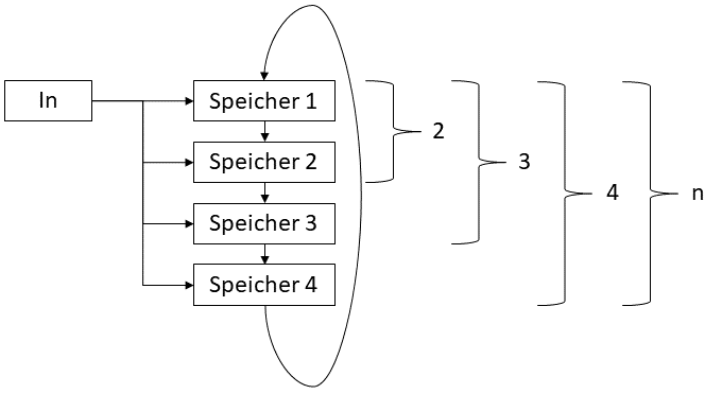
Der Min16 ist ein gewöhnlicher MIN-Baustein, wie jener aus der kitControl Palette, verfügt jedoch über 16 Eingänge.

2.6. Max16

Der Max16 ist ein gewöhnlicher MAX-Baustein, wie jener aus der kitControl Palette, verfügt jedoch über 16 Eingänge.

2.7. AverageValues

Der AverageValues ist ein Baustein zur Bildung eines Durchschnitts gespeicherter Werte. Die Update Period gibt an in welchem Intervall der Wert am In Slot eingelesen und gespeichert werden soll. Ist der Update Intervall 0 wird der Durchschnitt bei jeder Wertänderung des Eingangswerts gebildet. Der Slot Number Of Values gibt an wie viele Werte gespeichert werden sollen. Gebildet wird der Durchschnitt immer nur über die gespeicherten Werte.



Der Ausgang wird jeweils nach jedem Intervall neu berechnet und gesetzt.

2.8. BooleanLastChanged

Der BooleanLastChange ist ein Baustein, welcher über vier In Slots und einen Out Slot verfügt.

Der Baustein gibt am Out Slot den Wert des In Slots aus, welcher sich zuletzt geändert hat.

«Null» an einem Eingang wird nicht als Wert gewertet.

2.9. NumericLastChange

Der NumericLastChange ist ein Baustein, welcher über vier In Slots und einen Out Slot verfügt.

Der Baustein gibt am Out Slot den Wert des In Slots aus, welcher sich zuletzt geändert hat.

«Null» an einem Eingang wird nicht als Wert gewertet.

2.10. EnumLastChange

Der EnumLastChange ist ein Baustein, welcher über vier In Slots und einen Out Slot verfügt.

Der Baustein gibt am Out Slot den Wert des In Slots aus, welcher sich zuletzt geändert hat.

«Null» an einem Eingang wird nicht als Wert gewertet.

2.11. StringLastChange

Der StringLastChange ist ein Baustein, welcher über vier In Slots und einen Out Slot verfügt.

Der Baustein gibt am Out Slot den Wert des In Slots aus, welcher sich zuletzt geändert hat.

«Null» an einem Eingang wird nicht als Wert gewertet.

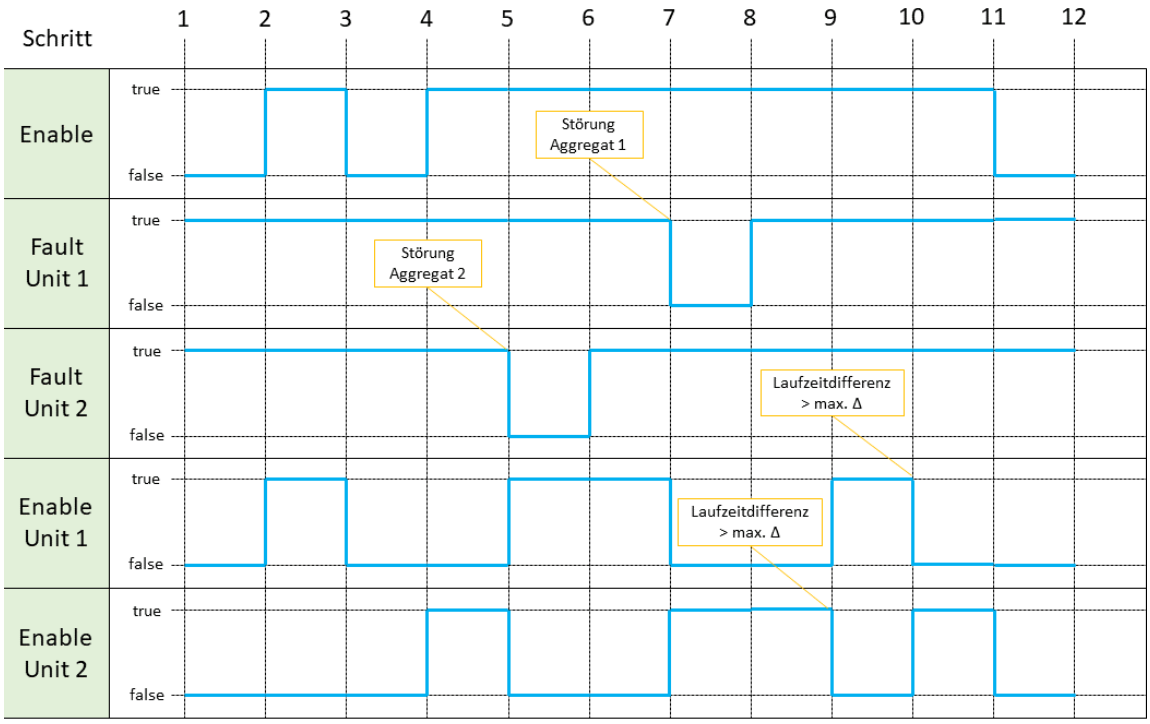
3. HVACLogic

3.1. DewPointCalculator

Der Taupunktrechner berechnet die absolute Luftfeuchtigkeit und den Taupunkt aufgrund der Temperatur und relativen Luftfeuchtigkeit. Die Berechnung wird immer bei einer Wertänderung am Temperatur Eingang oder am Eingang der relativen Feuchtigkeit neu durchgeführt.

3.2. RuntimeSwitch

Der RuntimeSwitch ist ein Baustein zur Laufzeit- und Störumschaltung zweier Aggregate. Bei einer Freigabe, wird immer zuerst das Aggregat mit der kleineren Laufzeit eingeschaltet. Ist die Differenz zwischen der Laufzeit des Laufenden Aggregats und der Laufzeit des stehenden Aggregats grösser als das maximale Delta, werden die Aggregate umgeschaltet.



Hat ein laufendes Aggregat eine Störung, so übernimmt das zweite Aggregat, unabhängig von seiner Laufzeit. Hat ein nicht laufendes Aggregat eine Störung, so wird die Laufzeitumschaltung nicht durchgeführt, auch wenn die Bedingungen dafür erreicht wären. Haben beide Aggregate eine Störung, so wird der Ausgang Total Failure auf true gesetzt.

3.3. OperatingMode

Der OperatingMode Baustein kann einen Temperaturwert über eine einstellbare Zeit Mitteln und danach aufgrund von Schwellwerten eine Betriebsart ausgeben. Zur Wertmittelung werden 288 Werte aufgezeichnet, was bei 24 Stunden einen interval von 5 min. entspricht.

Wird beim Aussentemperatur Eingang der Status «null» festgestellt, werden die Werte nicht mehr in die Durchschnittswerteberechnung miteinbezogen. Signalisiert wird dieser Zustand am Ausgang der durchschnittlichen Aussentemperatur durch den Status «stale». Sobald der Wert am Eingang nicht mehr den Status «null» hat, wird der Durchschnittswert wieder ordnungsgemäss berechnet und der Status am Ausgang ändert sich wieder auf «ok».

Die Ausgänge werden wie folgt gesetzt:

(durchschnittliche Aussentemperatur > Kühllimite Ein) & Heizen Aus = Kühlen Ein

durchschnittliche Aussentemperatur < Kühllimite Aus = Kühlen Aus

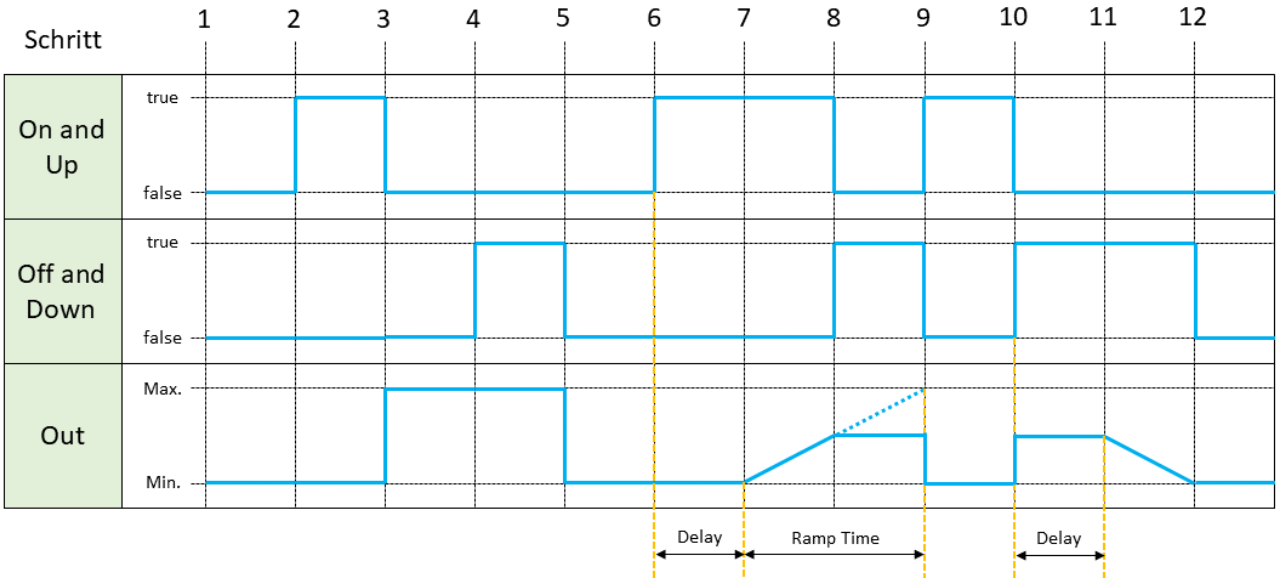
(durchschnittliche Aussentemperatur < Heizlimite Ein) & Kühlen Aus = Heizen Ein

durchschnittliche Aussentemperatur > Heizlimite Aus = Heizen Aus

Durch diese vergleiche entsteht automatisch ein Totband z.B. zwischen Heizen Aus und Kühlen Ein. Dieses Totband wurde mit absicht so einprogrammiert.

3.4. Dimmer

Der Dimmer ist ein Baustein, welcher bei Licht-Anwendungen zum Einsatz kommen kann. Die Funktion ist wie folgt:



Aufgepasst: Das Erkennen eines digitalen Eingangs unter 300ms ist mit Niagara schwierig, da alle Eingänge über Busprotokolle mit unterschiedlicher Geschwindigkeit eingelesen werden. Aus diesem Grund ist bei der Bewertung der Funktion des Dimmers gewisse Vorsicht geboten.

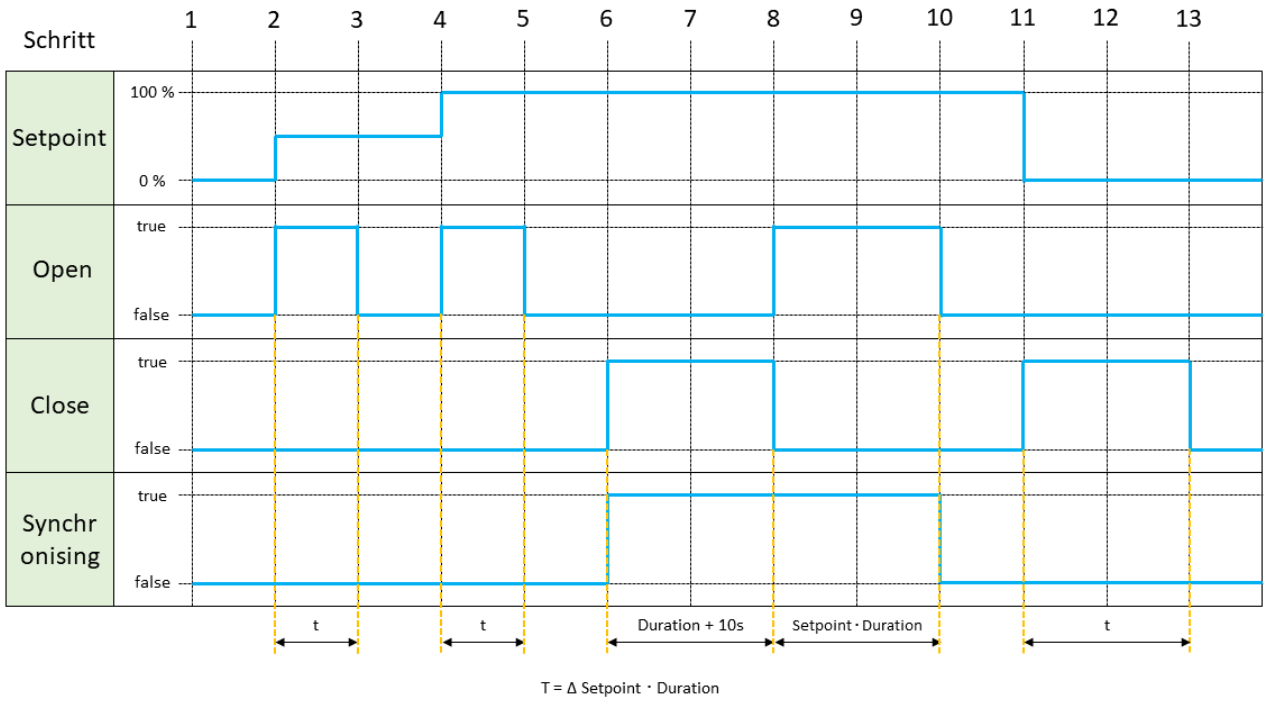
3.5. ThermalEnergyMeter

Der ThermalEnergyMeter Baustein kann die Thermische Energie in hydraulischen Anwendungen berechnen. So kann er die Aufgabe eines fehlenden Zählers übernehmen. Dafür benötigt werden lediglich die Vorlauf- und Rücklauftemperaturen sowie der Durchfluss. Über das «Enabled» Flag kann der Zähler Ein- bzw.- Ausgeschaltet werden.

Aufgrund des Messorts und der dort gemessenen Temperatur (Vorlauf- oder Rücklauf) wird die Dichte des Mediums und die daraus resultierende Wärmekapazität berechnet. Das Medium bezieht sich grundsätzlich auf Wasser. Mit dem «GlycolContent» kann der Glykolanteil angegeben werden, falls es sich um eine Kälteanwendung handelt. Der Glykolanteil hat einen Einfluss auf die Dichte und die Wärmekapazität.

3.6. ThreePointDrive

Der ThreePointDrive ist ein Baustein zur Ansteuerung von Drei-Punkt Antrieben. Die Funktion ist wie folgt:



Wird die Synchronisation eingeschaltet, wird im Intervall der Synchronisation der Antrieb voll geschlossen (+10s Sicherheit) und dann wieder gemäss dem Setpoint aufgefahren. Dies wird dazu genutzt, um die sich im Laufe der Zeit summierende Ungenauigkeit auszugleichen.

3.7. SwitchingHysteresis

Der Baustein SwitchingHysteresis stellt eine Schalthysterese dar. Diese wird z.B. Eingesetzt, um ein Heiz- oder Kühlbetrieb unter- bzw. oberhalb eines gewissen Schwellwerts ein- bzw. auszuschalten.

Der Slot Out wird auf true gesetzt, wenn der Wert an Slot inA minus den Wert von HysteresisOn grösser als der Wert von inB ist.

Der Slot Out wird auf false gesetzt, wenn der Wert an Slot inA plus den Wert von HysteresisOff kleiner als der Wert von inB ist.

3.8. PlantSwitch

Der Baustein PlantSwitch repräsentiert ein Anlagenschalter, an welchem ein Hardware- sowie Software-Anlagenschalter angeschlossen werden kann. Die Eingänge für den SoftwareAnlagenschalter haben nur dann einen Effekt auf die Ausgänge, wenn die Hardware-Anlagenschalter Eingänge den Status null haben oder der Hardwareanlagenschalter auf Auto steht. Falls mehrere Eingänge gleichzeitig anstehend sind, hat Aus immer Priorität.

3.9. VavSetpointCalc

Der VavSetpointCalc kann die Berechnung von verschobenen Sollwerten machen. Dies dient dazu, die in der VAV eingestellten Vmin/Vmax Werte neu zu setzen bzw. den relativen Sollwert darin neu zu berechnen. Somit kann die Einstellung von Vmin bzw. Vmax der VAV in den Baustein ausgelagert werden und bei Bedarf ohne Zugang zur VAV neu «eingestellt» werden. Zusätzlich kann die VAV auch als statischer Vorregler genutzt und auf einen fixen Volumensollwert geregelt werden. Die Messungen der Soll-Volumenströme vor Ort, entfallen.

Der Slot «Relative Setpoint» kann mittels Link oder via Action «Set Rel Setpoint» gesetzt werden. Anhang der Eingestellten Vmin/Vmax Werte (Ist Vmin und Vmax = 0, gelten 0 und Vnom) sowie den «Volumetric Flow Setpoint Facets» min und max Werte, wird dann der «Out» als relativer Sollwert berechnet.

Der «Volumetric Flow Setpoint» kann mittels Link oder via Action «Set Volume Flow Setpoint» gesetzt werden. Dann wird der «Out» als relativer Setpoint ebenfalls aufgrund der eingestellten Parameter berechnet. Dabei gelten ebenfalls die Regeln wie oben beschrieben.

4. Converters

4.1. DecToHex

Der Baustein DecToHex kann Dezimalzahlen zu Hexadezimalzahlen konvertieren. Der Input ist numerisch, der Ausgang ein String.

4.2. HexToDec

Der HexToDec Baustein konvertiert einen hexadezimalen Eingang, formatiert als String in eine numerische Dezimalzahl.

4.3. IntBitsToFloat

Im Baustein IntBitsToFloat werden Integer Eingangswerte am Ausgang als Float ausgegeben.

4.4. BitsToNumericDemux

Der Baustein BitsToNumericDemux kann bis zu 64 einzelne Bits als Dezimalzahl am numerischen Ausgang ausgeben. Dieser Baustein hilft z.B. ModBus Register zu beschreiben.

Die Anzahl gewünschter Bit Eingänge kann an der Property «NumberOfInputBits» bearbeitet werden.

4.5. VolumetricFlowToVolume

Der VolumetricFlowToVolume Baustein kann einen Durchfluss während einer Zeit als Volumen ausgeben und kumulieren. Diese Funktion kann z.B. dann hilfreich sein, wenn kein Zähler zur Verfügung steht, der Durchfluss bei eingeschalteter Pumpe jedoch bekannt ist.

4.6. PowerToEnergy

Der Baustein PowerToEnergy kann eine anstehende Leistung in kW aufgrund der Zeit zu einem Energiewert umrechnen und kumuliert speichern und ausgeben.

4.7. StringToAsciiInt

Der Baustein StringToAsciiInt kann String Zeichen als ASCII Code in einen Integer umwandeln. Dabei nimmer er immer zwei Zeichen, speichert das erste Zeichen in den ersten 8 Bit des Integer und das zweite Zeichen in den zweiten 8 Bit.

4.8. AsciiIntToString

Der Baustein AsciiIntToString kann bis zu 64 ASCII Code Integer Werte in einen String umwandeln. Dabei sind immer die ersten 8 Bit sowie die letzten 8Bit eines Integer ein ASCII Zeichen.

4.9. UnitConverter

Der UnitConverter Baustein kann Werte zwischen Einheiten umwandeln. Beispielsweise zwischen l/s und m^3/h. Es können jedoch nur Einheiten mit der gleichen Basis umgewandelt werden.

5. Util

5.1. JSON

5.1.1. GetJsonObject

Mit der Komponente GetJsonObject kann anhand eines definierten Key ein JSON-Object aus einem String extrahiert werden. Diese Komponente kann z.B. dafür verwendet werden, um eine API-Response aufzuschlüsseln und an einen spezifischen Wert zu gelangen. Der Wert wird als String beim Out Slot ausgegeben. Dank des Niagara Frameworks und dessen Kompatibilität, kann jedoch z.B. ein Wert «true» auf einen BooleanWritable verknüpft werden und wird auch als boolscher Wert interpretiert.

5.1.2. MakeSimpleJsonObject

Mit der Komponente MakeSimpleJsonObject kann aus einem Niagara Datentyp ein JSON-Object erstellt werden, welches am Out Slot ausgegeben wird. Mithilfe des Slots UseEnumOrdinal kann ausgewählt werden, ob der Enum Wert als String (Tag) oder als Integer (Ordinal) in das JSON-Object überführt wird.

Hinweis: Ein Wert wird nur in das JSON-Object übernommen, wenn sein Status gültig ist. Sind mehrere Werte und Status gelichzeitig gültig, so wird nur der letzte in der Reihenfolge von oben in das JSON-Object übernommen.

5.2. AlarmRoutingFilter

Der Baustein AlarmRoutingFilter kann als Filter z.B. zwischen einer Alarmklasse und einem Alarmempfänger eingesetzt werden. Dabei routet der Filter nur Alarme weiter, welche die Kriterien des Filter erfüllen.

Slotname	Beschreibung	Type
Time Range	Zeitbereich in welchem Alarme weitergeleitet werden	BTimeRange
Days Of Week	Wochentage an welchen Alarme weitergeleitet werden	BDaysOfWeekBits
Transitions	Alarmtransitionen welche weitergeleitet werden	BAlarmTransitionBits
Route Acks	Wert um die Weiterleitung von Quittierungen zu steuern	BBoolean
Alarm Filter	Filter um die Weiterleitung der Alarme zu steuern	BAlarmFilter

Alarm Filter Einträge können über ein Flag aktiviert oder deaktiviert werden. Dies setzt den Filter z.B. ausser Kraft, ohne dessen Einstellungen zu ändern.

5.2.1. Alarm Properties

Key	Beschreibung	Typ	Beispielwert
timestamp	Zeitstempel	AbsTime	20.04.2026 17:00
uuid	Alarm UUID	String	0000-0000…
sourceState	Quellzustand	String	Gekommen
ackState	Quittierungszustand	String	Unquittiert
ackRequired	Quittierung erforderlich	Boolean	true
source	Quelle	String	
alarmClass	Alarmklasse	String	defaultAlarmClass
priority	Priorität	Integer	0…255
normalTime	Zeit Gegangen	AbsTime	20.04.2026 17:00
ackTime	Zeit Quittiert	AbsTime	20.04.2026 17:00
user	Benutzer	String	Unknown User
alarmData	Siehe Alarmdaten	String	
alarmValue	Alarmwert	Boolean, Double, DynamicEnum etc.	true/50,0
lastUpdate	Letzte Aktualisierung	AbsTime	20.04.2026 17:00

5.2.2. Alarmdaten

Die Alarmdaten ist eine spezielle Property, welche frei im Filter konfiguriert werden kann. Die Alarmdaten eines Alarms enthalten jedoch sehr viele Informationen, weshalb folgend die wichtigsten Keys aufgelistet sind.

Key	Beschreibung	Typ	Beispielwert
alarmValue	Alarmwert	Boolean, Double, DynamicEnum etc.	true/50,0
Count	Alarmzähler	Integer	42
deadband	Totband	String	1,0
escalated	Eskaliert	String	level1
faultValue	Fehler Wert	String	50,0
fromState	Von Zustand	String	offnormal
highLimit	Oberes Limit	String	50,0
lowLimit	Unteres Limit	String	-20,0
msgText	Nachricht	String	OK
notes	Notizen	String	Kommentar
numericValue	Numerischer Wert	Integer	0
offnormalValue	Abnormal Wert	String	true
presentValue	Aktueller Wert	String	false
sourceName	Quellenname	String	BooleanWritable
status	Status	String	{overridden} @ 1
timeDelay	Zeitverzögerung	RelTime	0 ms
timeDelayToNormal	Zeitverzögerung zum Zustand «normal»	RelTime	0 ms
TimeZone	Zeitzone	TimeZone	Europe/Berlin (+1/+2)
toState	Zum Zustand	String	normal

Weiter können in einer AlarmExtension eigene Key-Value-Pairs hinzugefügt werden, welche dann in den Alarmdaten erscheinen. Diese Key-Value-Pairs nennen sich auf der AlarmExtension «Meta Data». Nach den in den Metadaten hinzugefügten Keys kann wiederum in den Alarmdaten gefiltert werden.

Wichtig bei der Eingabe der Filter ist, dass der Key exakt der Schreibweise oben entspricht. Es mag als Fehler erscheinen, dass z.B. Count mit grossem Anfangsbuchstaben geschrieben ist, dem ist jedoch nicht so. Die Schreibweise der Keys in der obigen Tabelle stimmt exakt.

Weiter müssen numerische Werte welche mit einem String Filter hinzugefügt werden, mit einem «,» als Trennzeichen (amerikanische Schreibweise) getrennt werden.

Wie die einzelnen Werte geschrieben werden müssen, ist am besten ersichtlich, wenn ein Alarm geöffnet wird und die Alarmdaten angeschaut werden. Die Schreibweise muss exakt mit jener übereinstimmen.

5.3. AlarmToString

Mit der Komponente AlarmToString können Alarme in einen String umgewandelt werden. Der Alarm wird zum Zeitpunkt des Auftretens in einen JSON formatierten String umgewandelt und am Out Slot ausgegeben.

Die JSON-Keys orientieren sich dabei an den gültigen Alarm-Properties. Diese sind im Kapitel Alarm Properties zu finden. Die Alarmdaten werden im JSON-Object «alarmData» mit den spezifizierten Keys ausgegeben.

Hinweis: Sämtliche Zeitstempel werden zu Unix Seconds gewandelt und als Integer in das JSON geschrieben. Dies stellt die Interoperabilität sicher.


```
{
  "ackTime":0,
  "alarmClass":"Prio_1",
  "alarmData":{
    "fromState":"normal",
    "escalated":"",
    "Count":1,
    "presentValue":"60,0",
    "toState":"highLimit",
    "TimeZone":"Europe/Berlin (+1/+2)",
    "alarmValue":60,
    "offnormalValue":"60,0",
    "msgText":"",
    "deadband":"0,0",
    "timeDelay":"0 ms",
    "company":"IBSYS",
    "location":"Winterthur",
    "sourceName":"NumericWritable",
    "highLimit":"50,0",
    "timeDelayToNormal":"0 ms",
    "status":"{ok} @ def"
  },
  "source":"local:|station:|slot:/Services/AlarmService/NumericWritable/OutOfRangeAlarmExt",
  "priority":255,
  "uuid":"874bdf45-d714-44e6-ad71-79697bec871e",
  "ackState":"Unquittiert",
  "normalTime":0,
  "alarmValue":60,
  "lastUpdate":1776697570,
  "ackRequired":true,
  "sourceState":"Gekommen",
  "user":"Unknown User",
  "timestamp":1776697570
}
```


5.4. StringReplace

Der Baustein StringReplace nimmt einen BStatusString am Slot „In“ entgegen, sucht nach String Zeichen des Slots „Search“ und ersetzt diese durch die Zeichen im Slot „Replace“. Dann gibt er den neuen String am Slot „Out“ aus.

5.5. HistoryCsvExporter

Mit dem HistoryCsvExporter Komponent, können Histories in eine .csv Dateien exportiert werden. Folgend werden die Slots und deren Funktion beschrieben:

Slotname	Beschreibung	Type
Status	Status des Bausteins	Status
Fault Cause	Fehlergrund im Fehlerfall	String
Make File	Wenn diese Option ausgeschaltet ist, wird das File nur am Topic Slot File ausgegeben und keine Datei generiert	Boolean
Directory	Pfad unter welchem der Export abgelegt wird	Ord
File Name	Dateiname der .csv Datei	Format
Append Date Time	Anfügen des aktuellen Zeitstempels an den Dateinamen	Boolean
Append Date Time Format	Formatierung des aktuellen Zeitstempels welcher an den Dateinamen angehängt wird	String
Target	Resultierender Pfad und Dateiname des Exports	String
Trigger	Auslöser für den Export	TimeTrigger
Last Export	Zeitpunkt des letzten erfolgreichen Exports	AbsTime
Include Header	Spaltentitel in den Export einbeziehen	Boolean
Export Props	Properties welche in die Datei exportiert werden sollen	HistoryCsvExportStructure
History Ords	Histories von welchen die Daten exportiert werden sollen	OrdList
Time Range	Zeitbereich welcher im Export enthalten sein soll	DynamicTimeRange
Date Format	Datumsformat für die Datums-Spalte im Export	String
Time Format	Zeitformat für die Zeit-Spalte im Export	String
Date Time Format	Zeitformat für die Zeitstempel-Spalte im Export	String
Value Format	Werteformat für die Werte-Spalte im Export	String
Csv Delimiter	.csv Trennzeichen	String
File	File Ausgang zum versenden via Email (ibsysAddOns)	Topic(Blob)

5.5.1. Properties

Bei den exportierten Properties handelt es sich, bis auf vier Ausnahmen, um History Properties.

History Properties:

- History Name
- Date
- Time
- Date Time
- Value
- Unit
- Trend Flags
- Status

Parent Component Properties:

- User Property 1
- User Property 2
- User Property 3
- User Property 4

Diese Properties verweisen auf ein Property Slot auf dem Parent der History Extension.

Damit die Properties auf dem Parent erkannt werden, müssen diese exakt folgendermassen benannt sein: «userProperty1», «userProperty2», «userProperty3» bzw. «userProperty4».

Der displayName der Properties darf, für eine bessere Beschriftung, abweichen. Damit ist es möglich, weitere Eigenschaften, welche als statische Information auf dem Punkt vorhanden sind, in den Export aufzunehmen.

Die Spaltennamen können frei überschrieben werden, als Standard wird der Propertyname verwendet. Dies geschieht mit dem Eintrag in das String-Eingabefeld (Column Name) neben der Property.

Date Time Format

Der Zeitstempel kann frei formatiert werden. Dabei gelten folgende Repräsentationen:

YYYY: Jahr in vier stellen (z.B. 2024)
MM: Monat in zwei Stellen (z.B. 07 für Juli)
MMM: Monat in langer Schreibweise (z.B. Juli)
DD: Tag in zwei Stellen (z.B. 22)
HH: Stunden im 24h Format in zwei Stellen (z.B. 16 für 16Uhr)
hh: Stunden im 12h Format in zwei Stellen (z.B. 4 für 16Uhr)
mm: Minuten in zwei Stellen
ss: Sekunden werden im Datum als Dateinamen-Anhang nicht dargestellt

Als Trennzeichen empfiehlt sich ein Unterstrich, Leerzeichen sowie ein paar weitere Zeichen werden nicht akzeptiert, da sie gegen die Dateinamen-Konvention verstossen.

Wichtig: Der Inhalt der User Properties wird zum Zeitpunkt des Exports für alle Zeilen in den Export geschrieben.

Hinweis: Existieren mehrere Trend Flags oder Status, werden diese mit einem Komma voneinander getrennt. Es empfiehlt sich als Delimiter ein Semikolon zu verwenden oder die Trend Flags und den Status ggf. nicht zu exportieren.

Hinweis Topic(Blob): Am Topic Slot wird das csv File nur als byte[] (Blob) weitergegeben, das Dateiformat, Filename usw. haben darauf keinen Einfluss und sind darin nicht enthalten.

5.6. BooleanChanged

Der BooleanChanged Baustein gibt bei einer Änderung am Eingang während einer einstellbare Zeit ein true am Ausgang aus. Eine Statusänderung wird nicht als Wertänderung gewertet.

5.7. NumericChanged

Der NumericChanged Baustein gibt bei einer Änderung am Eingang während einer einstellbare Zeit ein true am Ausgang aus. Eine Statusänderung wird nicht als Wertänderung gewertet.

5.8. EnumChanged

Der EnumChanged Baustein gibt bei einer Änderung am Eingang während einer einstellbare Zeit ein true am Ausgang aus. Eine Statusänderung wird nicht als Wertänderung gewertet.

5.9. StringChanged

Der StringChanged Baustein gibt bei einer Änderung am Eingang während einer einstellbare Zeit ein true am Ausgang aus. Eine Statusänderung wird nicht als Wertänderung gewertet.

5.1. StringBase64Encoder

Der StringBase64Encoder Baustein kann einen String, welcher am Input Slot anliegt, Base64 codieren und am Out Slot ausgeben. Der Input wird dabei zuerst in UTF-8 Bytes gewandelt und danach Base64 encodiert.

6. License Credits

Falls hier nichts anderes angegeben ist, benötigt jede Instanz einer Komponente ein License Credit.

Komponente	Benötigte License Credits (pro Instanz)
DewPointCalculator	100
RuntimeSwitch	100
ThermalEnergyMeter	100
AlarmRoutingFilter	500
HistoryCsvExporter	10’000