

# **KIS.ME**

Keep it simple. Manage everything.

# **KIS.MANAGER**

## **KPI-Handbuch**

Version 1.1, 07/2021

## Inhaltsverzeichnis

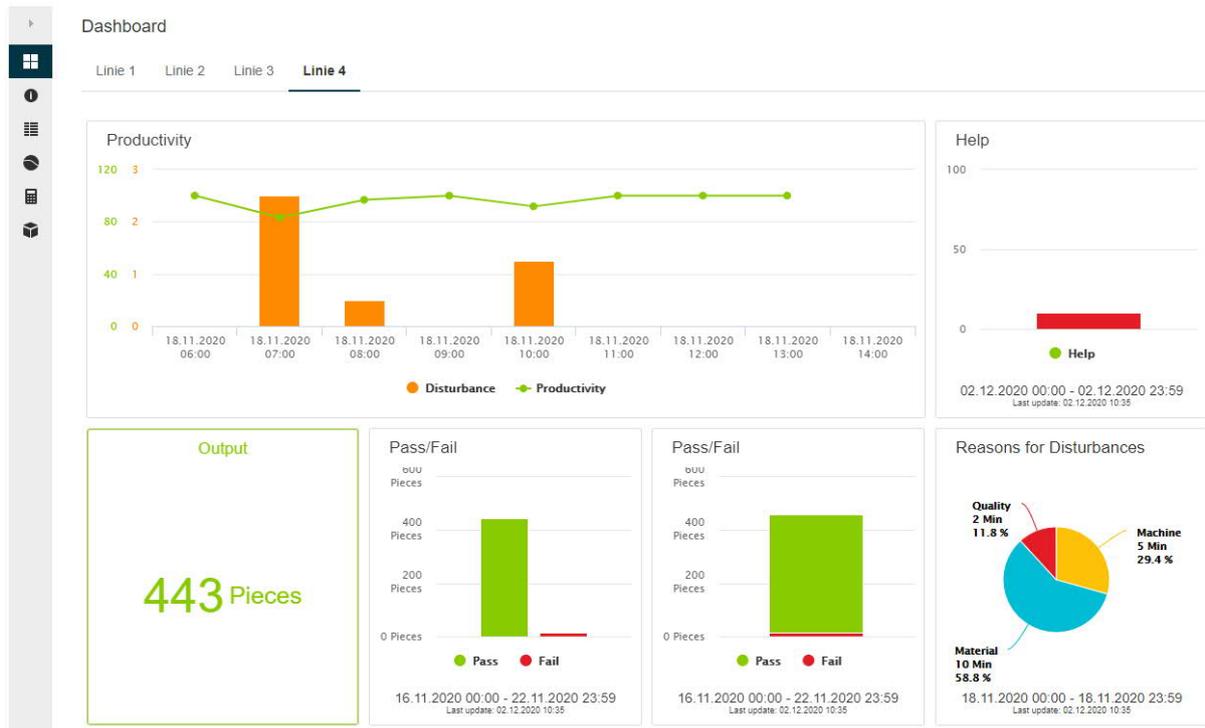
1	Einführung .....	3
2	Datenpunkte .....	4
2.1	Datenpunkte als Grundlagen für Kennzahlen.....	4
2.2	Datenpunkte als Arbeits-Erleichterung .....	5
3	Der Weg von Datenpunkten zu Kennzahlen.....	7
4	Grundsätzliches zur Formelsprache FLEX .....	8
5	Berechnungsmaske .....	10
5.1	Berechneter Datenpunkt anlegen.....	11
5.2	Kennzahlen-Berechnungen anlegen .....	12
6	Darstellung mittels Dashboard-Widgets .....	15
6.1	Diagramm-Formen .....	15
6.2	Konfigurations-Maske.....	17
7	Praxis-Tipps.....	21
8	Anhang .....	23
8.1	Tabelle mit typischen Auswertungen von Datenpunkten.....	23
8.2	Tabelle zur Übersetzung von LED-Farben in Zahlen .....	24
8.3	Operatoren der FLEX-Sprache.....	25
8.4	Beispiel-Formeln KPIs.....	27
9	Impressum.....	29

## 1 Einführung

Kennzahlen sind eine elementare Säule um Prozesse analysieren und beurteilen zu können. Sie schaffen notwendige Transparenz und ermöglichen gezielte Handlungen.

Der KIS.MANAGER bietet dazu attraktive Möglichkeiten. Die KIS.Devices (KIS.BOX, KIS.LIGHT, etc.) sammeln Daten vor Ort, im KIS.MANAGER werden sie verarbeitet und dargestellt.

Das Ergebnis kann zum Beispiel so aussehen:



Unterschiedliche Widgets stehen zur Darstellung von Kennzahlen auf den verschiedenen Dashboards des KIS.MANAGERs zur Verfügung.

Das folgende Handbuch erklärt den Weg vom Datenpunkt bis zur Kennzahl und lässt dabei auch die vielfältigen Darstellungsmöglichkeiten nicht aus.

## 2 Datenpunkte

Die Datenpunkte bilden den Input für jede Kennzahlen-Berechnung.

Bei einem Datenpunkt handelt es sich um einen Austauschwert zwischen KIS.Device und KIS.MANAGER. Unterschieden wird zwischen:

- ➔ Zustands-Daten (Button gedrückt, LED-Farbe grün, EINGANG high, ...)
- ➔ Meta-Daten (Firmware-Version, Hardware-Version, WLAN-Signalstärke, ...)

Über die Datenpunkte findet also die komplette Kommunikation zwischen KIS.Device und KIS.MANAGER statt.

Für die Berechnung von Kennzahlen sind die Zustands-Daten relevant. Sie transportieren die Prozess-Informationen, welche durch die Kennzahlen aufbereitet werden sollen.

### 2.1 Datenpunkte als Grundlagen für Kennzahlen

Der KIS.MANAGER bietet für jedes KIS.Device eine vollständige Übersicht zu den ausgetauschten Datenpunkten an:



Name	Type	Einheit	Datentyp	Aktueller Wert	Aktueller Zeitstempel
pin3Mode	Datenpunkt	-	Text	input	27.06.2021 20:47:18 (+02:00)
pin3Status	Datenpunkt	-	Boolescher Wert	false	28.06.2021 10:08:59 (+02:00)
pin4Frequency	Datenpunkt	-	Long	0	28.06.2021 10:08:59 (+02:00)
pin4Mode	Datenpunkt	-	Text	input	27.06.2021 20:47:18 (+02:00)
pin4Status	Datenpunkt	-	Boolescher Wert	false	28.06.2021 10:08:59 (+02:00)
serialNumber	Datenpunkt	-	Text	A000R0000000	27.06.2021 20:47:15 (+02:00)
subnet	Datenpunkt	-	Text	255.255.255.0	27.06.2021 20:47:15 (+02:00)
wifiChannel	Datenpunkt	-	Text	6 (2437 MHz)	27.06.2021 20:47:15 (+02:00)
wifiSignalStrength	Datenpunkt	-	Text	-61 dBm	27.06.2021 20:47:15 (+02:00)
wifiSsid	Datenpunkt	-	Text	RAFI_JOT	27.06.2021 20:47:15 (+02:00)

Grundsätzlich stehen Datenpunkte des folgenden Datentyps zur Auswertung zur Verfügung:

- ➔ Double - Gleitkommazahl (z.B.: 1,234)
- ➔ Long - Ganze Zahl (z.B.: 1)
- ➔ Boolean - Logischer Wert (z.B.: true)

Eine Übersicht zu den auswertbaren Datenpunkten findet sich im Anhang in Kapitel 8.1

## 2.2 Datenpunkte als Arbeits-Erleichterung

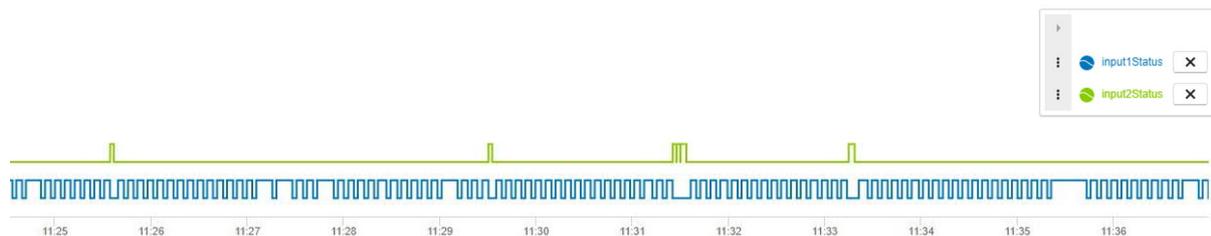
Die Übersicht über die Datenpunkte eines Assets erleichtert die Arbeit bei der Erstellung von KPIs und die Kontrolle ob das errechnete Ergebnis plausibel ist.

In der Übersicht kann:

- Nach auswertbaren Datenpunkten gefiltert werden
- Der aktuelle Wert eines/mehrerer Datenpunkte angeschaut werden
- Der zeitliche Verlauf eines/mehrerer Datenpunkte angeschaut werden
- Die Historie eines/mehrerer Datenpunkte exportiert werden
- Die Historie eines Datenpunkts gelöscht werden (Löschen-Zeitraum einstellbar)

Bevor mit einer Auswertung begonnen wird, ist es sinnvoll sich den Daten-Strom anzuschauen der über das KIS.Device an den KIS.MANAGER geliefert wird.

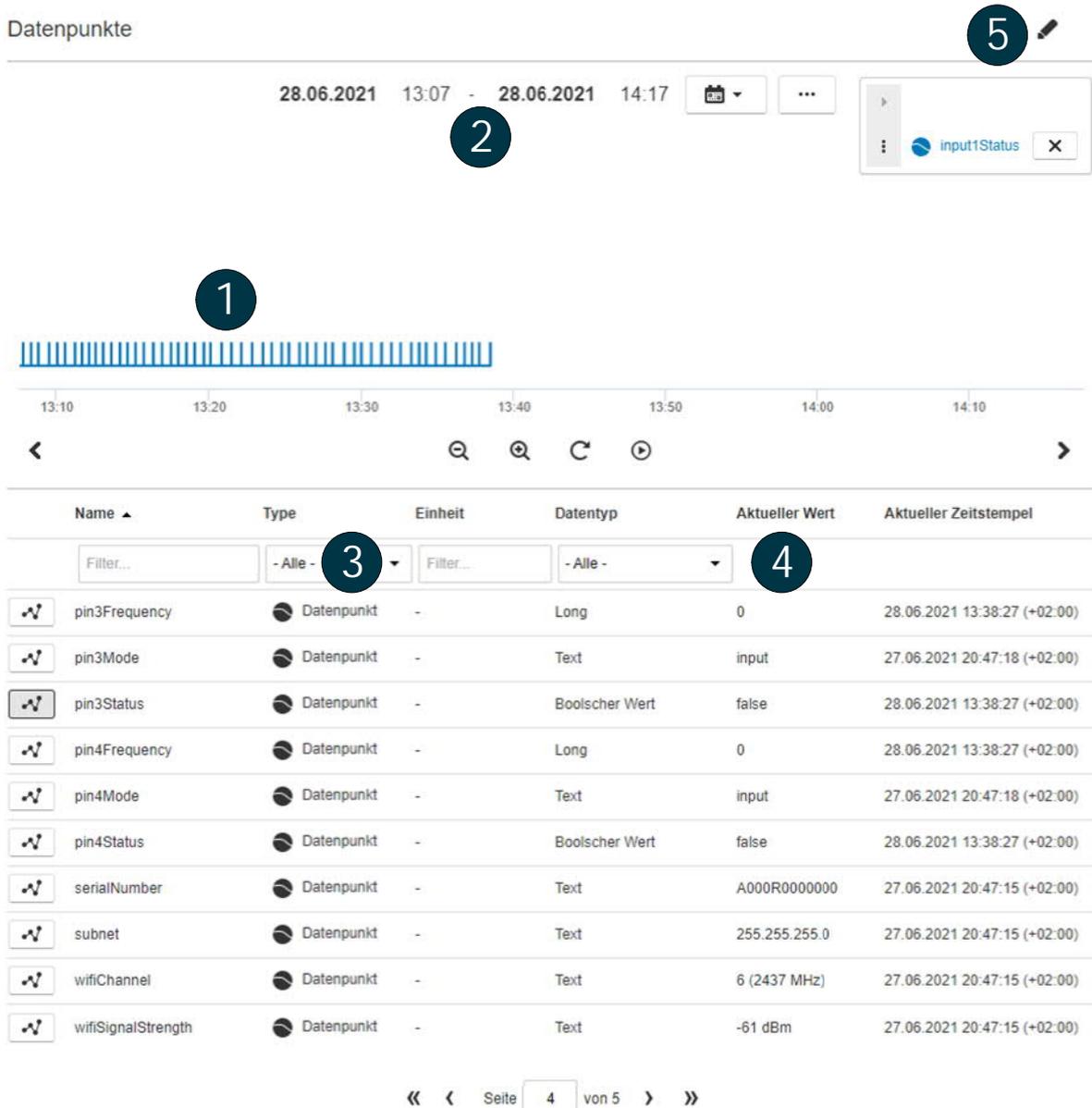
In diesem Beispiel ist ein Sensor mit Input 1 eines KIS.Devices verbunden und ein anderer Sensor mit Input 2. Wenn in der Datenpunkt-Übersicht die beiden Datenpunkte input1Status und input2Status ausgewählt werden, kann der zeitliche Verlauf wie folgt aussehen:



Dieser Blick auf die empfangenen Roh-Werte hilft, um das Berechnungs-Ergebnis später auf Plausibilität zu prüfen. Außerdem kann erkannt werden in welchem Datenbereich sich die Rohdaten bewegen.



Grundsätzlich ist es wichtig zu wissen, dass Datenpunkte event-getrieben aktualisiert werden. Das bedeutet, ein KIS.Device schickt eine Aktualisierung-Nachricht, wenn sich etwas verändert hat.

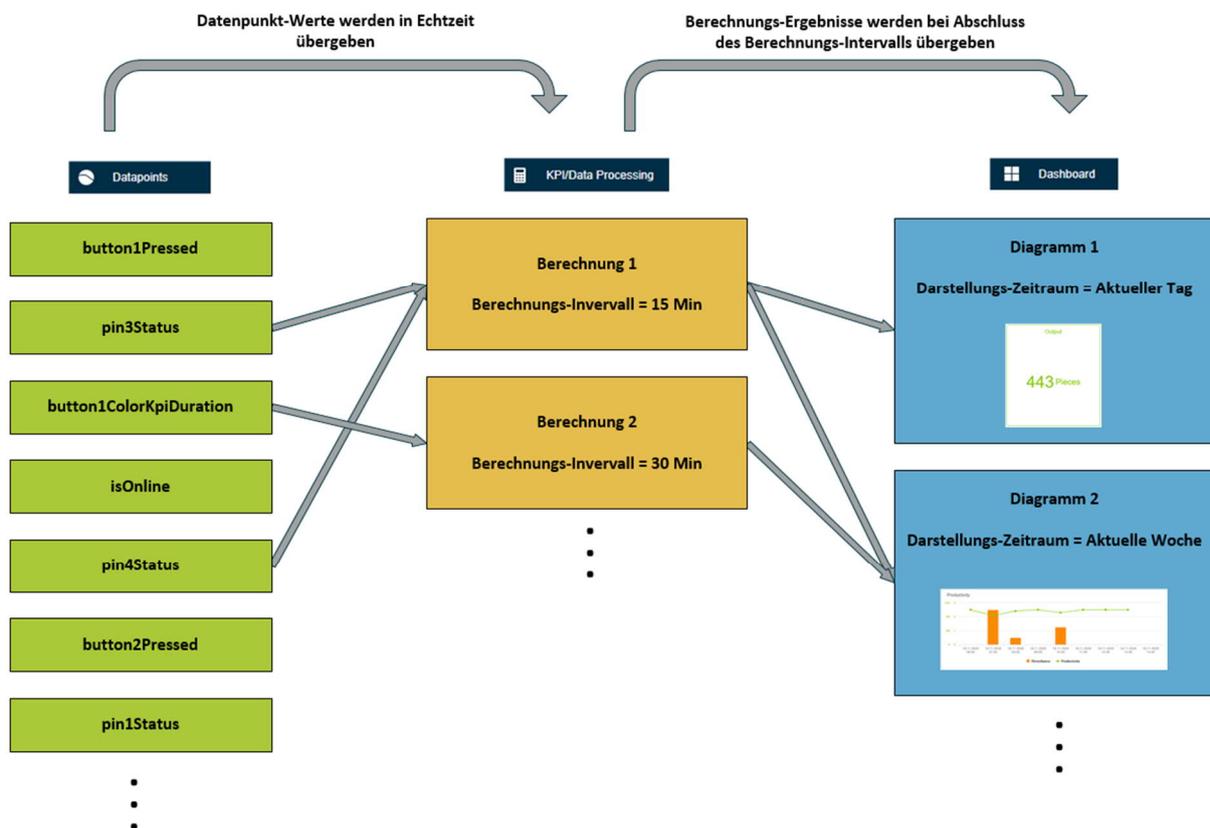


- 1 Zeitlicher Verlauf eines ausgewählten Datenpunkts
- 2 Einstellung des Darstellungs- bzw. Export-Zeitraums → Export über ...
- 3 Datenpunkt oder Berechneter Datenpunkt → siehe Kapitel 5.1
- 4 Aktueller Wert eines Datenpunkts
- 5 Datenpunkte löschen bzw. Datenpunkt-Historie löschen

## 3 Der Weg von Datenpunkten zu Kennzahlen

Um aus den Werten eines/mehrerer Datenpunkte eine Kennzahl zu berechnen, werden drei Stadien durchlaufen:

- ➔ Roh-Werte: Diese werden aus den Datenpunkten gewonnen (siehe Kapitel 2)
- ➔ Vorverarbeitung: Datenpunkte werden mittels Formeln zu Kennzahlen verrechnet (siehe Kapitel 5)
- ➔ Darstellung: Berechnungs-Ergebnisse werden aggregiert und dargestellt (siehe Kapitel 6)



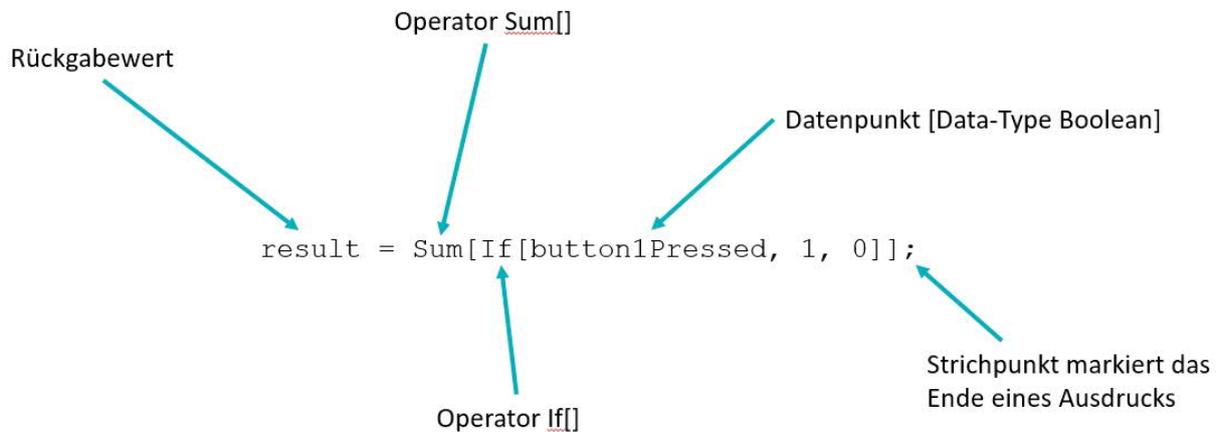
## 4 Grundsätzliches zur Formelsprache FLEX

Flex bietet eine Ausdruckssprache zum Auswerten, Aggregieren und Bearbeiten von Zeitreihendaten.

Bedeutung im Zusammenhang mit dem KIS.MANAGER:

Die permanent anfallenden Datenpunkt-Werte können über die FLEX-Sprache miteinander verglichen, verrechnet, zusammengefasst, bearbeitet, ... werden um als Ergebnis eine gewünschte Kennzahl zu erhalten.

Beispiel – Zähle wie oft Button 1 gedrückt wurde:



- Der Operator If[] liefert jedes Mal eine 1, wenn der Datenpunkt button1Pressed den Wert „true“ annimmt → Button gedrückt
- Der Operator If[] liefert jedes Mal eine 0, wenn der Datenpunkt button1Pressed den Wert „false“ annimmt → Button nicht gedrückt
- Der Operator Sum[] summiert die Ergebnisse des Operators If[] über die eingestellte Zeitspanne auf
- Der Rückgabewert enthält das Ergebnis der Berechnung welches zur Darstellung übergeben werden kann



- Jede Zeile einer Berechnung wird mit einem Strichpunkt abgeschlossen
- Leerzeichen können problemlos an allen Stellen eingefügt werden
- Der Rückgabewert kann frei benannt werden, Sonderzeichen ausgenommen

Grundsätzliche Möglichkeiten zur Datenpunkt-Verrechnung mit KIS.MANAGER (ausführliche Definition im Anhang unter 8.3):

## 1. Numerische Operationen

+ , - , * , / , ^	Plus, Minus, Mal, Geteilt, Hoch
Round	Rundung auf nächste ganze Zahl
Abs	Bildet den Betrag der Zahl

## 2. Logische Operationen

And	Dieser Wert UND dieser Wert
Or	Dieser Wert UND/ODER dieser Wert
Xor	ENTWEDER dieser Wert ODER dieser Wert
Not	NICHT dieser Wert

## 3. Daten-Aggregation

Sum	Summe
Mean	Mittelwert
Count	Zustandszähler
Max , Min	Maximum , Minimum
First , Last	Erster bzw. Letzter Wert einer Periode
Stdev	Standardabweichung
RisingEdge	Anzahl steigender Flanken (digitaler Wert)
FallingEdge	Anzahl fallender Flanken (digitaler Wert)

## 4. Vergleichende Operationen

> , <	Größer, Kleiner
>=, <=	Größer gleich, Kleiner gleich
==	Ist gleich
!=	Ist ungleich

## 5. Sonstiges Operationen

If	Wenn, dann, sonst
Duration	Dauer eines Zustands in Millisekunden
Filter	Unplausible Daten herausfiltern
Counter	Differenz zwischen jedem Wert-Wechsel

## 5 Berechnungsmaske

Die Berechnungen können jeweils für jedes Asset angelegt werden. Unter folgendem Link wird der Einstieg zu den Berechnungsmasken erreicht:



- 1 Liste bereits angelegter Berechnungen
- 2 Klick auf **+** öffnet Optionen um Berechnungen hinzuzufügen
- 3 Leere Berechnungsmaske öffnet sich
- 4 Liste der bereits angelegten Berechnungen aller Assets öffnet sich. Bei Auswahl einer dieser Berechnungen wird ein (nicht verknüpftes) DUPLIKAT erstellt  
→ Details unter Kapitel 7
- 5 Liste der bereits angelegten Berechnungen aller Assets öffnet sich. Bei Auswahl einer dieser Berechnungen wird eine VERKNÜPFUNG erstellt  
→ Details unter Kapitel 7
- 6 Teilen von Berechnungen → Die Berechnung kann mit einer Asset Group geteilt werden → Jedes Asset in der Asset Group wird mit der Berechnung verknüpft  
→ Details unter Kapitel 7
- 7 Löschen von Berechnungen

## 5.1 Berechneter Datenpunkt anlegen

Die Funktion „Berechneter Datenpunkt“ dient dazu, vorhandene Datenpunkte umzurechnen bzw. in anderer Form zur Verfügung zu stellen.

Beispiel: Ein Datenpunkt ist als Boolean vom Device verfügbar und soll in einen Zahlenwert überführt werden

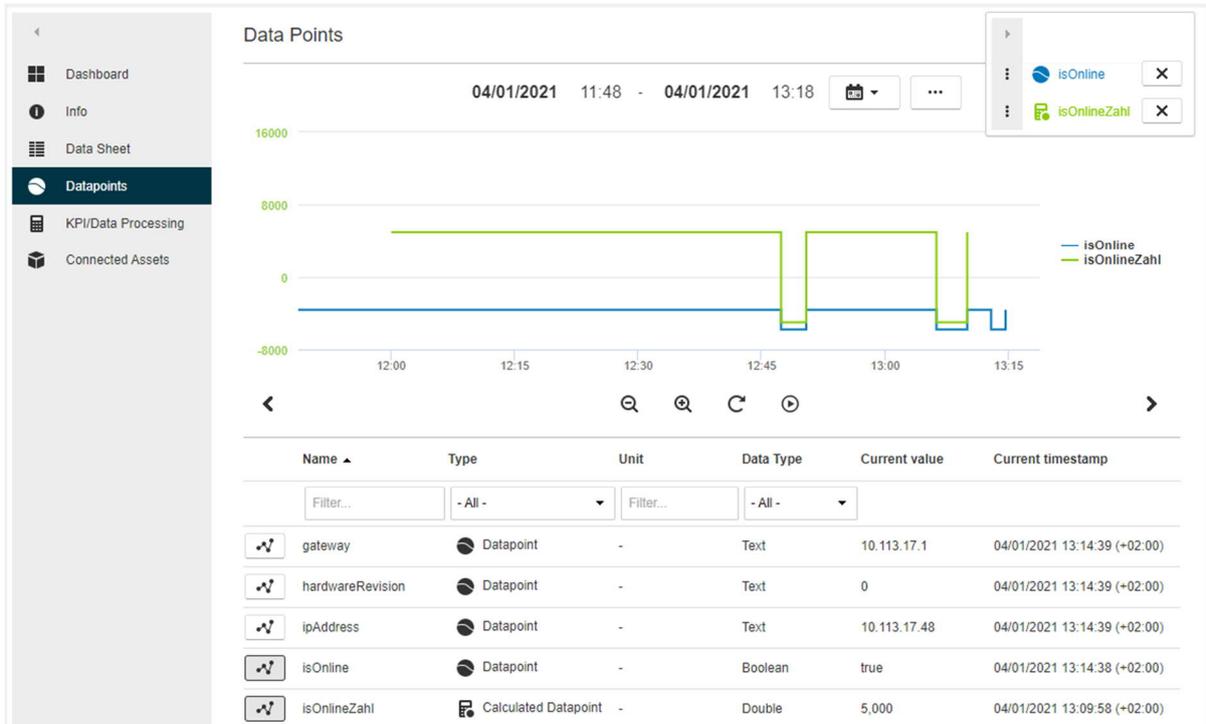
### 5.1.1 Anlegen eines Beispiels

The screenshot shows the 'Data Processing Definition editieren' interface. It includes a sidebar with navigation icons, a main form area, and a bottom bar with 'Speichern' and 'Abbrechen' buttons. The form is divided into sections: '1. Name:', 'Veröffentlichen als ...', '2. Datenpunkte:', '3. Berechnung:', and '4. Hilfsfunktionen:'. The '1. Name:' field contains 'OnlineZahl'. The 'Veröffentlichen als ...' section has 'Berechneter Datenpunkt' selected. The '2. Datenpunkte:' section has 'input\_0' selected in the 'ist verbunden' field. The '3. Berechnung:' section contains the formula '1 OnlineZahl = If[input\_0, 5000, -5000];'. The '4. Hilfsfunktionen:' section is currently empty.

- 1 Unter diesem Namen erscheint der berechnete Datenpunkt in der Liste der Datenpunkte
- 2 Auswahl auf „Berechneter Datenpunkt“
- 3 Input-Datenpunkt auswählen
- 4 Die Berechnung wird immer direkt ausgeführt wenn der Input-Datenpunkt sich verändert → keine Aggregations-Periode wählbar

## 5.1.2 Ansicht in der Liste und Möglichkeit zur Weiterverarbeitung

Wie in der Datapoint-Übersicht ersichtlich ist, ist der neu berechnete Datenpunkt „isOnlineZahl“ in der Liste aufgetaucht. Dieser Datenpunkt kann nun wie jeder andere dargestellt, exportiert, gelöscht oder in KPI-Berechnungen verwendet werden.



## 5.2 Kennzahlen-Berechnungen anlegen

### 5.2.1 Abgrenzung zu berechneten Datenpunkten

Der große Unterschied zwischen berechneten Datenpunkten und KPIs ist die zeitliche Basis. Während berechnete Datenpunkte eine sofortige Umrechnung eines Datenpunkts ermöglichen, sind KPIs dazu vorgesehen, Daten über einen Zeitraum hinweg zu verarbeiten.

Dazu stehen bei KPIs sogenannte Verarbeitungszeiträume von 15 Minuten, 30 Minuten und 60 Minuten zur Verfügung. Der Verarbeitungszeitraum definiert, in welchem Zyklus ein Ergebnis zur Verfügung steht. Das Ergebnis wiederum basiert auf den Datenpunkt-Werten, welche innerhalb des Verarbeitungszeitraums angefallen sind.

Einige Beispiel-Berechnungen sind im Anhang unter Kapitel 8.4 einsehbar.

## 5.2.2 Anlegen eines Beispiels

The screenshot shows the 'Data Processing Definition editor' interface. It includes a sidebar with navigation icons, a main workspace with numbered callouts (1-6), and a bottom bar with 'Speichern' and 'Abbrechen' buttons. The workspace contains:
 

- 1. Name:** 'Zähle Taster-Betätigungen'. Radio buttons for 'Berechneter Datenpunkt' and 'KPI' (selected).
- 2. Datenpunkte:** 'button1Pressed' with 'input\_0' selected. A '+ Variable hinzufügen' button is present.
- 3. Berechnung:** A code editor with '1 result = RisingEdge[input\_0];'. A 'Definition ist in Ordnung' status is shown.
- 4. Hilfsfunktionen:** A section for helper functions.
- 5. Output:** 'Ausgabewert: result' and 'Ausgaben-Vorschau'.
- 6. Initial value:** A toggle switch for 'Initial value'.

- 1 Auswahl auf KPI
- 2 Input-Datenpunkt(e) auswählen, welche für die Berechnung notwendig sind
- 3 Verarbeitungszeitraum festlegen und den gewünschten Startzeitpunkt der Berechnung einstellen
- 4 Wenn ein bereits anliegender Datenpunkt-Wert aus einer Vor-Periode ignoriert werden soll, Initial value deaktivieren
- 5 Umfasst die Berechnung mehrere Teilergebnisse → Rückgabewert explizit auswählen
- 6 Über Ausgaben-Vorschau kontrollieren ob die Berechnung das gewünschte Ergebnis liefert

### 5.2.3 Besonderheiten



Wird eine Berechnung abgespeichert und später wieder geöffnet, so wird der Startzeitpunkt auf das aktuelle Datum gesetzt. Ist das nicht gewünscht, muss es manuell wieder auf den richtigen Startzeitpunkt geändert werden. Wurde die Maske nur zur Ansicht geöffnet und ohne zu speichern wieder verlassen, ist diese Anpassung nicht notwendig. Es gilt immer noch das Datum, welches beim ersten Speichern eingestellt war



Datenpunkte vom Datentyp „Text“ können zwar als Input-Datenpunkt ausgewählt werden, aber die Formelsprache kann deren Inhalt nicht verarbeiten. Verarbeitbar sind die Datenpunkte mit dem Datentyp „Boolean“, „Long“ und „Double“



Wird der Name einer Berechnung nachträglich geändert und abgespeichert, so müssen die Widgets, in denen die Berechnung verwendet wird nochmals neu konfiguriert werden. Ansonsten zeigen sie den Fehler: „The KPI(s) with the name `xxx` can not be loaded – please assure your configuration“

## 6 Darstellung mittels Dashboard-Widgets

Nun da die KPIs berechnet werden, fehlt noch die Darstellung auf den Dashboards. Die Berechnungen werden separat für jedes Asset angelegt. Zur Darstellung stehen sie zum einen auf ihren Asset-Dashboards zur Verfügung, aber auch auf dem Dashboard einer Asset Group wenn das Asset einer solchen angehört.

Navigation:



oder:



### 6.1 Diagramm-Formen

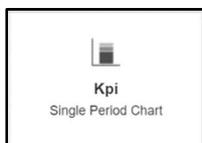
#### 6.1.1 Single-Value



Darstellung eines einfachen numerischen Werts:



#### 6.1.2 Single Period Chart



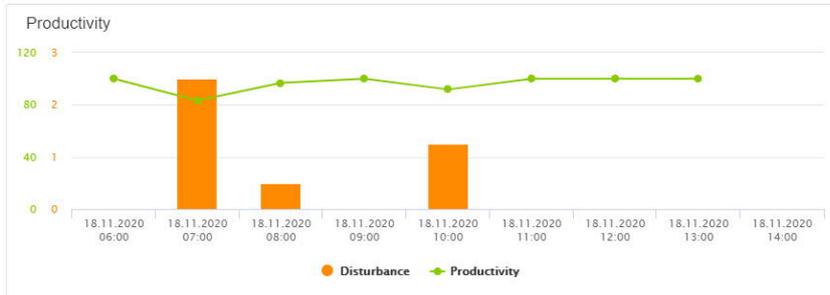
Darstellung eines gestapelten Säulendiagramms:



## 6.1.3 Aggregated Chart



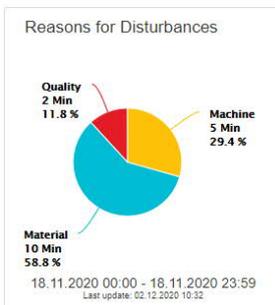
Kombination von Säulen- und Liniendiagrammen:



## 6.1.4 Pie Chart



Darstellung eines Kreisdiagramms:



## 6.1.5 Single Value Column



Darstellung eines Säulendiagramms (Schwellwerte für Farb-Umschläge einstellbar):



## 6.2 Konfigurations-Maske

Einstellungen ändern

Überschrift **1**

Aggregationstyp **2** Einheit  KPIs stapeln  Stacked chart **3**

Zeitfenster **4**

12.07.2021 00:00 - 19.07.2021 00:00

Arbeitsschicht **5**

Mon Tue Wed Thu Fri Sat Sun von 08:00 bis 16:00

**6**

- 1** Überschrift frei wählbar
- 2** Aggregationstyp festlegen (Details siehe Kapitel 6.2.2)
- 3** Diagramme stapeln oder nebeneinander abbilden
- 4** Darstellungszeitraum wählen (Details siehe Kapitel 6.2.1)
- 5** Darstellungszeitraum durch Arbeitsschichten verfeinern
- 6** Auswahl welche KPI-Berechnung(-en) mittels dieses Widgets dargestellt werden soll(-en)

## 6.2.1 Darstellungszeitraum

Der gewählte Zeitraum entscheidet darüber, welche der verfügbaren Berechnungsergebnisse in die Darstellung einfließen.

Wie der Grafik zu entnehmen ist, kann zwischen einigen vordefinierten Zeiträumen gewählt werden. Das kann beispielsweise der heutige Tag, die letzte Woche oder der vergangene Monat sein. Wird ein Zeitraum in der Kategorie „Letzter“ oder „Vorheriger“ ausgewählt, kann außerdem entschieden werden wie viele vergangene Stunden/Tage/Wochen/Monate betrachtet werden:

Welcher Zeitraum ausgewählt wurde, zeigt die Datumsangabe unter dem Auswahl-Feld. Der Zeitraum verändert sich dynamisch mit fortschreitender Zeit (Ausnahme ist das Zeitfenster „Benutzerdefiniert“).

Der Schiebe-Regler „Arbeitschicht“ ermöglicht es, Arbeitstage und Arbeitszeiten in die Spezifizierung des Darstellung-Zeitraum miteinzubeziehen.

Im Beispiel werden ausschließlich Berechnungs-Ergebnisse von Montag bis Freitag zwischen 8 Uhr und 16 Uhr der aktuellen Woche bei der Darstellung beachtet.

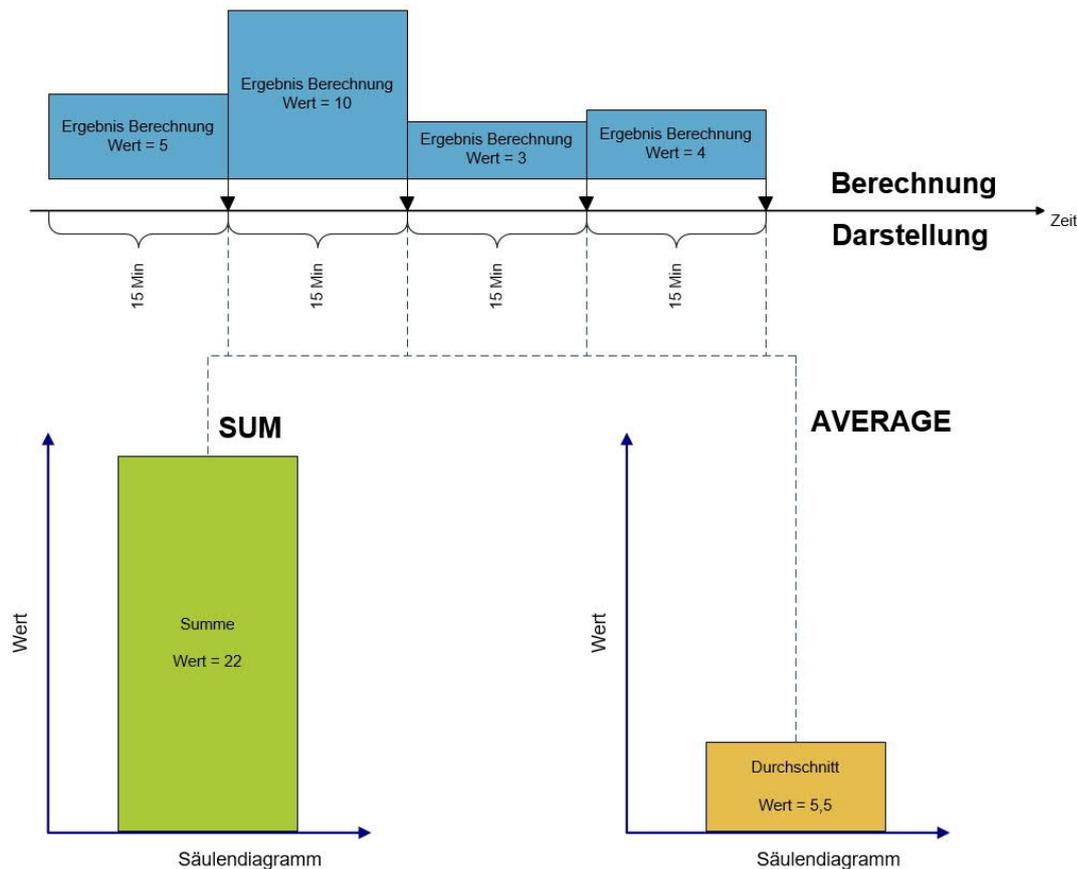
## 6.2.2 Aggregation

Wie in Kapitel 3 beschrieben, findet bei der Berechnung der KPIs eine Vor-Aggregation statt über einen Zeitraum von 15, 30 oder 60 Minuten. Dies wird bereits beim Anlegen der KPI-Definition (Berechnung) festgelegt. Für die Darstellung wiederum wurde nun konkret der Zeitraum ausgewählt, der für den User interessant ist. Nun muss der User noch entscheiden, wie die Berechnungs-Ergebnisse in diesem Zeitraum dargestellt werden sollen.

Zur Verfügung stehen folgende Aggregations-Typen:

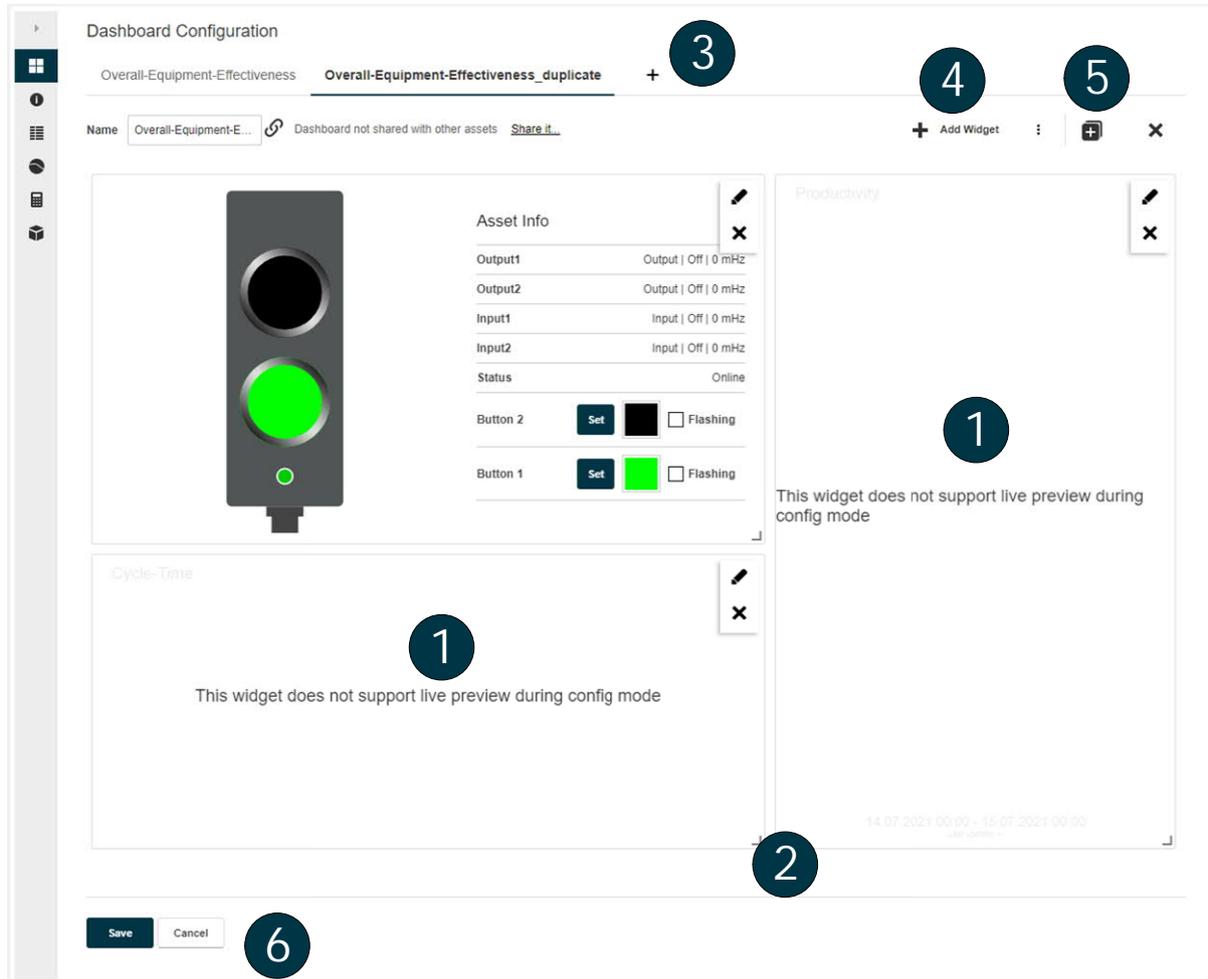
- ➔ SUM - Summe
- ➔ AVERAGE - Durchschnitt
- ➔ MAX - Maximum
- ➔ MIN - Minimum

Beispiel – Darstellungszeitraum 1 Stunde bei einer Berechnungs-Periode von 15 Minuten:



## 6.2.3 Dashboard einrichten

Ist das KPI-Widget konfiguriert, wird es auf dem Dashboard platziert. Hier gibt es noch einige Möglichkeiten um die Darstellung anzupassen:

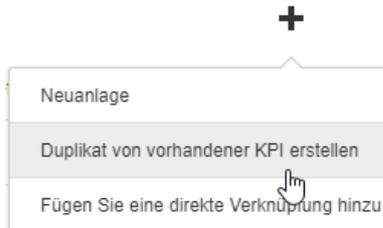


- 1 KPI-Widgets
- 2 Größe des Widgets verändern durch Klicken und Ziehen
- 3 Weiteres Dashboard hinzufügen
- 4 Weiteres Widget hinzufügen
- 5 Dashboard duplizieren
- 6 Änderungen am Dashboard speichern

## 7 Praxis-Tipps

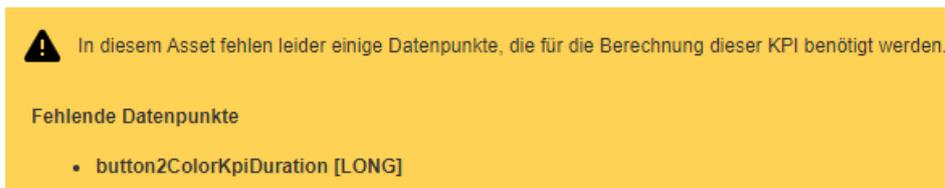
### → Berechnungen duplizieren

Soll eine Berechnung mehrfach verwendet werden, muss sie nicht jedes Mal komplett neu angelegt werden. Bestehende Berechnungen können dupliziert werden - auch von einem Asset zum anderen.



Die duplizierte Berechnung wird nach der Auswahl in der Bezeichnung mit `_duplicate` gekennzeichnet.

Falls in der Berechnung Datenpunkte verwendet werden, über die das Ziel-Asset nicht verfügt, so macht der KIS.MANAGER mit einer Warnmeldung darauf aufmerksam:



### → Berechnungen teilen

Während bei der Berechnungs-Duplizierung keine Abhängigkeit zwischen Quelle und Ziel besteht, wird beim Teilen die Berechnung mit einer kompletten Asset-Group geteilt und verknüpft:



Das Teilen hat zur Folge, dass die Berechnung in der Berechnungs-Übersicht jedes der Asset Group angehörigen Assets auftaucht. Dort kann sie jeweils auch zur Bearbeitung geöffnet werden. Da eine Verknüpfung besteht, wirkt sich eine Änderung global aus. Auch wenn die Berechnungen miteinander verknüpft sind, bezieht sich das nur auf die Formeln. Die Datenpunkt-Werte sind nach wie vor für jedes Asset individuell und somit steht für jedes Asset ein separates Berechnungs-Ergebnis zur Verfügung.

Das Teilen wird in der Berechnungs-Übersicht wieder beendet:



## → Datenpunkte löschen

Datenpunkte können teilweise oder ganz gelöscht werden. Teilweise bedeutet, dass die Historie in einem gewählten Zeitraum gelöscht wird.

Beispiel-Szenarien, wann die Löschung zum Einsatz kommt:

- Während der Inbetriebnahme sind unplausible Datenpunkt-Werte aufgetreten, die nicht in die Analyse einbezogen werden sollen
- Ein berechneter Datenpunkt wird nicht mehr benötigt und soll aus der Liste der Datenpunkte wieder entfernt werden



Wird versehentlich ein Datenpunkt gelöscht, der für Device-Funktionen wichtig ist, beeinträchtigt das nicht die Kommunikation zwischen KIS.MANAGER und KIS.Device. Die Historie des Datenpunkts ist zwar verloren, aber bei der nächsten Nachricht zwischen KIS.Device und KIS.MANAGER wird der betroffene Datenpunkt wieder neu hinzugefügt.

## 8 Anhang

### 8.1 Tabelle mit typischen Auswertungen von Datenpunkten

Die folgende Tabelle listet auf, welcher Datenpunkt für welche Auswertung typischerweise verwendet wird:

Datenpunkt	Auswertung	Datentyp	Besonderheit
button1Pressed button2Pressed	Anzahl Button-Presses	Boolean	- Aktualisiert sich alle 15 Minuten
input1Status input2Status	Anzahl Zustände Eingang Dauer Zustände Eingang	Boolean	
output1Status output2Status	Anzahl Zustände Ausgang Dauer Zustände Ausgang	Boolean	
button1ColorKpi button2ColorKpi led1ColorKpi	Anzahl Farb-Zustände	Long	- Jede Farbe hat eine definierte Zahl (siehe 8.2) - Aktualisiert sich alle 15 Minuten (Wert: 99)*
button1ColorKpiDuration button2ColorKpiDuration led1ColorKpiDuration	Dauer Farb-Zustände	Long	- Jede Farbe hat eine definierte Zahl (siehe siehe 8.2) - Aktualisiert sich alle 15 Minuten
isOnline	Dauer Online-Zustände	Boolean	

\*Die Datenpunkt-Werte werden alle 15 Minuten aktualisiert, auch wenn keine Veränderung stattgefunden hat. Bei der Zählung von Farb-Zuständen kann das zu einer Verfälschung des Ergebnisses führen. Liegt zum Beispiel der Farb-Zustand 0 (Blau) über längere Zeit an, so würde er bei jeder Datenpunkt-Aktualisierung erneut gezählt werden. Um das zu vermeiden, wird bei der zyklischen Aktualisierung der Wert 99 anstelle des Farb-Zustands eingetragen. Die 99 bedeutet der Farb-Zustand liegt weiterhin an, er triggert aber nicht die KPI-Berechnung erneut.

Wichtig: Das gilt nur für die Zählung der Farb-Zustände und die in der Tabelle aufgeführten Datenpunkte in der entsprechenden Zeile.

## 8.2 Tabelle zur Übersetzung von LED-Farben in Zahlen

Damit Farben in Analysen verwendet werden können, ist ihnen ein numerischer Wert zugewiesen worden. Die folgende Tabelle zeigt, welche Farbe zu welcher Zahl gehört:

Farbe	RGB-Hex-Code	Zahl
	Beispiel: Werte des Datenpunkts button1Color	Beispiel: Werte der Datenpunkte button1ColorKpi und button1ColorKpiDuration
Blau	#0000FF	0
Türkis	#00FFFF	1
Schwarz	#000000	2
Grün	#00FF00	3
Magenta	#FF00FF	4
Rot	#FF0000	5
Weiß	#FFFFFF	6
Gelb	#FFFF00	7

## 8.3 Operatoren der FLEX-Sprache

### 1. Numerische Operationen

Funktion	Beschreibung	Beispiel	Daten-Typ
Plus[] oder +	Plus	result = Plus[Var1, Var2] result = Var1 + Var2	Double Long
Time[] oder *	Mal	result = Time[Var1, Var2] result = Var1 * Var2	Double Long
Power[] oder ^	Hoch	result = Power[Var1, 2] result = Var1 ^ 2	Double Long
Round[]	Rundung auf nächste ganze Zahl	result = Round[Var1]	Double Long
Abs[]	Betrag	result = Abs[Var1]	Double Long

### 2. Logische Operationen

Funktion	Beschreibung	Beispiel	Daten-Typ
And[] oder &&	Und	result = And[Var1, Var2] result = Var1 && Var2	Boolean
Or[] oder	Und/Oder	result = Or[Var1, Var2] result = Var1    Var2	Boolean
Not[] oder !	Nicht	result = Not[Var1] result = !Var1	Boolean
Xor[]	Entweder oder	result = Xor[Var1, Var2]	Boolean

### 3. Vergleichende Operationen

Funktion	Beschreibung	Beispiel	Daten-Typ
Greater[] oder >	Größer	res = Greater[Var1, Var2] res = Var1 > Var2	Double Long
Less[] oder <	Kleiner	res = Less[Var1, Var2] res = Var1 < Var2	Double Long
Equal[] oder ==	Gleich	res = Equal[Var1, Var2] res = Var1 == Var2	Double Long
Unequal[] oder !	Ungleich	res = Unequal[Var1, Var2]	Double Long
GreaterEqual[] oder >=	Größer gleich	res = GreaterEqual[Var1, Var2] res = Var1 >= Var2	Double Long
LessEqual[] oder <=	Kleiner gleich	res = LessEqual[Var1, Var2] result = Var1 <= Var2	Double Long

#### 4. Daten-Aggregation (über definierten „Verarbeitungszeitraum“)

Funktion	Beschreibung	Beispiel	Daten-Typ
Sum[]	Summe numerischer Werte	result = Sum[Var1]	Double Long
Mean[]	Mittelwert numerischer Werte	result = Mean[Var1]	Double Long
Count[]	Anzahl der Dateneingaben	result = Count[Var1]	Boolean
Max[] oder Min[]	Maximum / Minimum numerischer Werte	result = Max[Var1] result = Min[Var1]	Double Long
First[] oder Last[]	Erster / letzter Wert einer Periode	result = First[Var1] result = Last[Var1]	Double Long
Stdev[]	Standardabweichung	result = Stdev[Var1]	Double Long
Percentil[]	Berechnet das definierte Perzentil	result = Percentil[Var1, 99]	Double Long
RisingEdge[]	Anzahl steigender Flanken	result = RisingEdge[Var1]	Boolean
FallingEdge[]	Anzahl fallender Flanken	result = FallingEdge[Var1]	Boolean

#### 5. Intervalle

Funktion	Beschreibung	Beispiel	Daten-Typ
Start[]	Gibt den Start-Timestamp eines Zustands zurück	result = Start[Var1]	Double Boolean Long
End[]	Gibt den End-Timestamp eines Zustands zurück	result = End[Var1]	Double Boolean Long
Duration[]	Gibt die Dauer eines Zustands in Millisekunden zurück	result = Duration[Var1]	Double Boolean Long
Interval[]	Gibt die Dauer der eingestellten Processing period in Millisekunden zurück	result = Interval[]	-

#### 6. Sonstige Funktionen

Funktion	Beschreibung	Beispiel	Daten-Typ
Counter[]	Gibt die Differenz zwischen jedem Wert-Wechsel zurück	result = Counter[Var1]	Double Long
Filter[]	Filtert unplausible Daten heraus	result = Filter[Var1 > 10 && Var1 < 100]	Double Long
If[]	Wenn, dann, sonst – basiert auf Wahrheits-Wert	result = If[Var1, 1, 0]	Boolean

## 8.4 Beispiel-Formeln KPIs

Bezeichnung	Datenpunkte	Formel	Beschreibung
DurationColor_KIS.BOX	button1ColorKpiDuration (as "led_button1")	Button1Blue = Round[Sum[If[led_button1 == 0, Duration[led_button1], 0]] / 60000];	Dauer Farbzustand in Minuten (in diesem Beispiel Blau)
DurationColor_KIS.LIGHT	led1ColorKpiDuration (as "led1")	LedBlue = Round[Sum[If[led1 == 0, Duration[led1], 0]] / 60000];	Dauer Farbzustand in Minuten (in diesem Beispiel Blau)
DurationOnline	isOnline (as "input_0")	DurationOnline = Round[Sum[If[input_0, Duration[input_0], 0]] / 60000];	Dauer Online-Zustand in Minuten
DurationOffline	isOnline (as "input_0")	DurationOffline = Round[Sum[If[Not[input_0], Duration[input_0], 0]] / 60000];	Dauer Offline-Zustand in Minuten
DurationOutput	output1Status (as "output1")	DurationOutput1 = Round[Sum[If[output1, Duration[output1], 0]] / 60000];	Dauer High-Zustand eines digitalen Ausgangs
DurationInput	input1Status (as "input1")	DurationInput1 = Round[Sum[If[input1, Duration[input1], 0]] / 60000];	Dauer High-Zustand eines digitalen Eingangs
CountButtonColor_KIS.BOX	button1ColorKpi (as "led_button1")	Button1Blue = Sum[If[led_button1 == 0, 1, 0]];	Anzahl Farb-Zustände (in diesem Beispiel Blau)
CountLedColor_KIS.LIGHT	led1ColorKpi (as "led1")	LedBlue = Sum[If[led1 == 0, 1, 0]];	Anzahl Farb-Zustände (in diesem Beispiel Blau)
CountButtonPresses	button1Pressed (as "button1")	Button1Pressed = Sum[If[button1, 1, 0]];	Anzahl Button-Gedrückt

Bezeichnung	Datenpunkte	Formel	Beschreibung
CountOutputHigh	output1Status (as "output1")	Output1High = RisingEdge[output1];	Anzahl High-Zustände eine digitalen Ausgangs
CountOutputLow	output1Status (as "output1")	Output1Low = FallingEdge[output1];	Anzahl Off-Zustände eine digitalen Ausgangs
CountInputHigh	input1Status (as "input1")	Input1High = RisingEdge[input1];	Anzahl High-Zustände eine digitalen Eingangs
CountInputLow	input1Status (as "input1")	Input1Low = FallingEdge[input1];	Anzahl Off-Zustände eine digitalen Eingangs
Example_FirstPassYield	input1Status (as "passed")	NumberPassed = RisingEdge[passed]; NumberFailed = RisingEdge[failed]; Output = NumberPassed + NumberFailed; FirstPassYield = Round[100 / Output * NumberPassed]; FirstPassYieldCorrected = Filter[FirstPassYield > 0];	Pin 3 zählt Gut-Teile  Pin 4 zählt Schlecht- Teile  Berechnung Verhältnis (First-Pass-Yield)
	input2Status (as "failed")		
Example_AverageCycleTime	input1Status (as "input1")	UpTimeInput1 = If[input1, Duration[input1], 0] / 1000; DownTimeInput1 = If[Not[input1], Duration[input1], 0] / 1000; ZyklusInput1 = Round[Mean[Filter[UpTimeInput1 > 0]] + Mean[Filter[DownTimeInput1 > 0]]];	Durchschnittliche Zyklus Zeit zwischen zwei steigenden Flanken
Example_Productivity	input1Status (as "input1")	LineCycle = RisingEdge[input1]; Productivity = Round[LineCycle / (Interval[] / 60000)];	Anzahl Teile pro Zeiteinheit
Example_Output	input1Status (as "input1")	LineCycleInput1 = RisingEdge[input1];	Ausbringung in Stück

## 9 Impressum

### **Herausgeber**

RAFI GmbH & Co. KG

Ravensburger Str. 128-134, D-88276 Berg / Ravensburg

Tel.: +49 751 89-0, Fax: +49 751 89-1300

[www.rafi.de](http://www.rafi.de), [info@rafi.de](mailto:info@rafi.de)

### **Copyright**

© RAFI GmbH & Co. KG, 2020. Alle Rechte vorbehalten.

Der Inhalt dieses Dokuments darf nicht ohne vorheriges schriftliches Einverständnis der RAFI GmbH & Co. KG reproduziert oder an andere Personen weitergegeben werden.

Die Verwendung von Handelsnamen, Markennamen usw. in diesem Dokument berechtigt nicht, diese Namen als frei zu betrachten.

### **Haftungsausschluss**

Es wurden alle notwendigen Schritte durchgeführt, um sicherzustellen, dass dieses Dokument vollständig und korrekt ist. Sollte es zu bestimmten Punkten trotzdem Fragen geben, wenden Sie sich bitte an die RAFI GmbH & Co. KG oder einen autorisierten Vertreter.

Die Informationen in diesem Dokument können ohne vorherige Ankündigung geändert werden und stellen keine Verpflichtung für die RAFI GmbH & Co. KG dar. RAFI GmbH & Co. KG übernimmt für mögliche Fehler in diesem Dokument keine Haftung.