



MOBA Mobile Automation AG

# Spezifikation

## *Weight()*

Version 2.000

<b>Produkt</b>	<b>MRW 4-20mA</b> (Momenten unabhängige Redundante Wägezelle)
<b>Auftraggeber</b>	<b>MOBA Mobile Automation AG</b> Kapellenstraße 15 65555 Limburg Germany
<b>Auftragnehmer</b>	<b>MOBA Mobile Automation AG</b> Kapellenstraße 15 65555 Limburg Germany

Dokument erstellt von	Datum	Unterschrift
M.Offenbach	12.05.2022	

Diese Dokumentation des Unittests basiert auf einem Vordruck der MOBA AG.

Der Inhalt darf ausschließlich den am Projekt beteiligten Personen zugänglich gemacht werden. Insbesondere die Weitergabe an Dritte ist ohne ausdrückliche schriftliche Erlaubnis der MOBA AG nicht erlaubt.

Außerhalb des gemeinsamen Projektes darf kein Teil dieser Unterlagen für irgendwelche Zwecke vervielfältigt oder übertragen werden, unabhängig davon, auf welche Art und Weise oder mit welchen Mitteln dies geschieht.

Die hier getroffenen Festlegungen schließen nicht aus, dass in einer gesonderten Geheimhaltungsvereinbarung weiterreichende oder abweichende Vereinbarungen zur Wahrung der Vertraulichkeit getroffen und festgeschrieben werden.

**Copyright by**

MOBA Mobile Automation AG

Kapellenstr. 15

D-65555 Limburg

Internet: [www.moba.de](http://www.moba.de)



## Inhaltsverzeichnis

1	Einführung.....	4
1.1	Vorwort.....	4
1.2	Änderungshistorie .....	4
1.3	Ansprechpartner.....	5
1.4	Anhänge.....	5
1.5	Glossar.....	5
2	Weight().....	6
2.1	Beschreibung .....	6
2.2	Spezifikation .....	7
3	Flowchart.....	10
4	Gegenüberstellung - Weight() - V1.103 und V2.000 .....	11
5	Kommentare.....	12
6	Anhang.....	13

# 1 Einführung

## 1.1 Vorwort

Die MOBA AG versteht sich als Partner für die Entwicklung und Lieferung kundenspezifischer Elektronikkomponenten und daraus zusammengestellter Steuerungssysteme, die für den Einsatz an mobilen Maschinen konzipiert sind.

Die hier vorliegende Spezifikation beschreibt das exakte Verhalten der Funktion *Weight()* der Datei *Weight.c*

Dies beginnt mit der Angabe der Übergabeparameter sowie dem Rückgabewert der Funktion.

Es folgen dann die Beschreibungen des Verhaltens der Funktion

Jede Beschreibung wird indiziert festgehalten. Somit ist in weiteren Dokumenten leicht Bezug auf die Spezifikation zu nehmen.

## 1.2 Änderungshistorie

Version	Datum	Kapitel	Änderung / Ergänzung
1.0	12.05.2022	alle	Erstellung

### 1.3 Ansprechpartner

**MOBA Mobile Automation AG**

Kapellenstraße 15

65555 Limburg

Name	Position	Telefonnummer	E-Mail
Boris Zils	Produktmanager	+49(0)6431-9577-123	<a href="mailto:b.zils@moba.de">b.zils@moba.de</a>
Sebastian Schlesies	Vertrieb	+49(0)6431-9577-267	<a href="mailto:s.schlesies@moba.de">s.schlesies@moba.de</a>
Jürgen Stiller	Entwicklungsleiter	+49(0)6431-9577-282	<a href="mailto:j.stiller@moba.de">j.stiller@moba.de</a>
Norbert Lipowski	Entwicklung	+49(0)6431-9577-137	<a href="mailto:n.lipowski@moba.de">n.lipowski@moba.de</a>

### 1.4 Anhänge

Dokumentname	Beschreibung

### 1.5 Glossar

Abkürzung / Fachbegriff	Beschreibung / Definition
MRW	Momenten unabhängige Redundante Wägezelle
DMS	Dehnungsmessstreifen

## 2 Weight()

### 2.1 Beschreibung

Diese Funktion umfasst alle Berechnungen und Verarbeitungen vom unbehandelten Rohmesswert des Analog-Digital-Wandlers bis hin zu dem normierten und von Temperatureinflüssen bereinigten Gewichtswert.

Die Reihenfolge der Berechnungen sieht dabei wie folgt aus:

- Wenn ein neuer Messwert vom ADC vorliegt diesen einlesen
- Rohmesswert filtern
- Nullpunkt vom gefilterten Rohmesswert abziehen
- Temperatur- und E-Modul-Kompensation durchführen
- Rohmesswert auf ‚kg‘ normieren
- Tara-, Netto- und Bruttogewicht ermitteln
- Untersuchung auf Gewichtsschwankung

Für fast alle Berechnungen nutzt man die Funktionalitäten der Bibliotheksfunktion ‚Measurement‘.

## 2.2 Spezifikation

Alle Spezifikationen sind in aufsteigender Reihenfolge zu erfüllen!

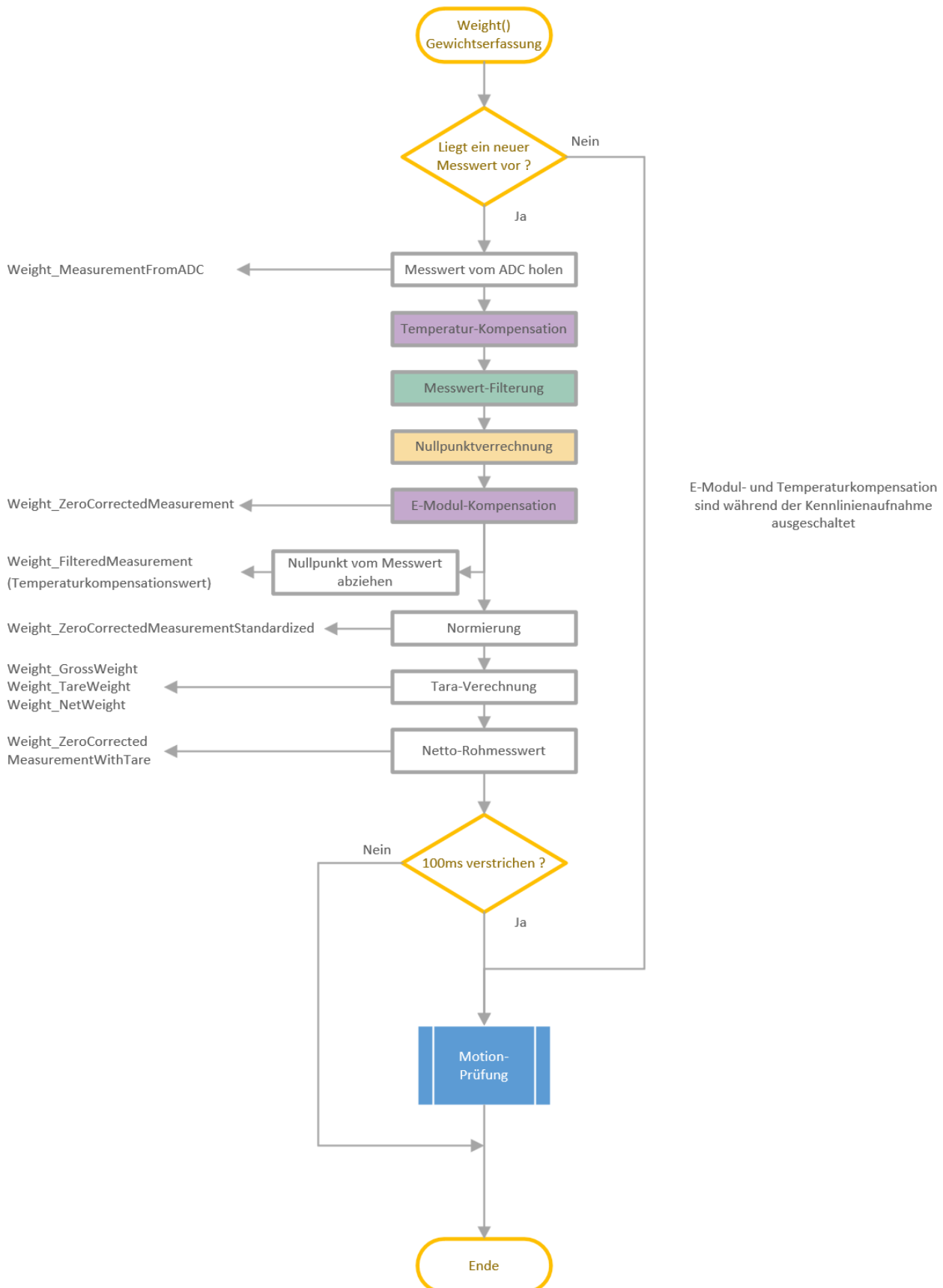
Weight()		
Index	Parameter	Datentyp
4.2.0.0	void	
Rückgabe		Datentyp
4.2.1.0	void	
Verhalten		Bemerkung
4.2.2.0	Zunächst ist zu prüfen, ob ein neuer Wandlungswert vorliegt ( <i>ADuC836_ADCIsNewConversionValue(ADuC836_ADC_PRIMARY) != 0</i> ) oder ob das Flag zur Gewichtssimulation gesetzt ist ( <i>g_SystemControl.bySimulate' &amp; 0x01</i> ).	Prüfung auf neuen Wandlungswert oder auf Simulation  <i>ADuC836_ADC_PRIMARY = 0</i>
4.2.2.1	<u>Keine der in 4.2.2.0 genannten Bedingungen trifft zu:</u> Nach Überprüfung auf stabilen Gewichtswert wird die Funktion verlassen	Es liegt kein neuer Messwert vor und keine Gewichtssimulation aktiviert.  Überprüfung auf stabilen Gewichtswert: s. Spezifikation 4.2.2.11
4.2.2.2	<u>Eine der in 4.2.2.0 genannten Bedingungen trifft zu:</u> Die Funktion <i>ADuC836_ADCGetConversionValue()</i> holt nun den aktuellen Messwert vom ADC ab und legt diesen in der globalen Union-Variablen <i>Measurement.nLong'</i> ab. Zuätzlich wird das Ergebnis in der globalen Variablen <i>Weight_MeasurementFromADC</i> festgehalten	<i>SYSTEM_CND_LEDS_4_DEBUG_P06_CHECK_WEIGHTING_CYCLE</i> ist nicht definiert <i>MIT_GEWICHTSSIMULATION</i> ist nicht definiert Gewichtswert vom ADC holen und ablegen. <i>ADuC836_ADC_PRIMARY = 0</i>
4.2.2.3	<u>Eine der in 4.2.2.0 genannten Bedingungen trifft zu:</u> Über den Aufruf der Bibliotheksfunktion <i>MRW_Compensation_TemperatureCompensation()</i> und der Übergabe der Adresse des bis zu diesem Zeitpunkt ermittelten Messwerts (Variable <i>Measurement.nLong'</i> ), diesen von den Temperatureinflüssen befreien.	Temperaturkompensation
4.2.2.4	<u>Eine der in 4.2.2.0 genannten Bedingungen trifft zu:</u> Über den Aufruf der Bibliotheksfunktion <i>Measurement_Processing()</i> und der Angabe von <i>MEASUREMENT_PROCESSING_FILTER</i> als ersten Parameter, wird die Filterung des erfassten Rohmesswerts durchgeführt.	Messwertfilterung  <i>MEASUREMENT_PROCESSING_FILTER = 4</i> <i>WEIGHT_WEIGHTCHANNEL = 0</i>
4.2.2.5	<u>Eine der in 4.2.2.0 genannten Bedingungen trifft zu:</u> Über den Aufruf der Bibliotheksfunktion <i>Measurement_Processing()</i> und der Angabe von <i>MEASUREMENT_PROCESSING_ZERO</i> als ersten Parameter, wird die Filterung des gefilterten Rohmesswerts durchgeführt.	Nullpunktverrechnung  <i>MEASUREMENT_PROCESSING_ZERO = 1</i> <i>WEIGHT_WEIGHTCHANNEL = 0</i>

4.2.2.6	<p><u>Eine der in 4.2.2.0 genannten Bedingungen trifft zu:</u>  Die Kompensation des E-Moduls ist nur zulässig, wenn diese aktiviert ist und keine Kennlinienaufnahme stattfindet.  Zur Feststellung der beiden Status, die Variable , Global.chEModulCompensationOn' abfragen (nicht aktiv: 0) und die Rückgabe der Funktionen <i>MRW_Compensation_GetRecCharacteristicsOnOffStatus()</i> auswerten (Kennlinienaufnahme nicht aktiv: 0)</p>	E-Modul-Kompensation
	<p><u>E-Modul-Kompensation darf ausgeführt werden:</u>  Den Einfluss der E-Moduls aus dem aktuellen Messwert gemäß der Formel:  <math display="block">MW = MW + (Temp - 20^{\circ}C) * E\text{-Modul-Konstante}</math> herausrechnen.</p>	<p><b>MW:</b> Messwert[digit] &lt;- Measurement.nLong  <b>Temp:</b> Prozessortemperatur[°C] &lt;- , Global.byTemperature'  <b>E-Modul-Konstante:</b> (-0.0005) &lt;- FACTOR_E_MODUL</p>
4.2.2.7	<p><u>Eine der in 4.2.2.0 genannten Bedingungen trifft zu:</u>  Aus Kompatibilitätsgründen zu früheren Versionen, ist dem kompensierten Messwert wieder der Nullpunkt hinzuzurechnen.  Hierzu über die Bibliotheksfunktion <i>Measurement_GetZero()</i> den Nullpunkt auslesen und dem kompensierten Messwert <i>Weight_ZeroCorrectedMeasurement</i> hinzuzaddieren. Das Ergebnis der Berechnung in der Globalen <i>Weight_FilteredMeasurement</i> ablegen.</p>	<p>Nullpunkt wieder hinzuaddieren</p> <p><math>WEIGHT\_WEIGHTCHANNEL = 0</math></p>
4.2.2.8	<p><u>Eine der in 4.2.2.0 genannten Bedingungen trifft zu:</u>  Über den Aufruf der Bibliotheksfunktion <i>Measurement_Processing()</i> und der Angabe von <i>MEASUREMENT_PROCESSING_STANDARDIZATION</i> als ersten Parameter, wird Normierung des kompensierten Rohmesswerts durchgeführt.</p>	<p>Normierung der Messwerts</p> <p><math>MEASUREMENT\_PROCESSING\_STANDARDIZATION = 2</math>  <math>WEIGHT\_WEIGHTCHANNEL = 0</math></p>
4.2.2.9	<p><u>Eine der in 4.2.2.0 genannten Bedingungen trifft zu:</u>  Über den Aufruf der Bibliotheksfunktion <i>Measurement_Processing()</i> und der Angabe von <i>MEASUREMENT_PROCESSING_TARE</i> als ersten Parameter, wird die Berechnung der Gewichtswerte anhand eines vorab hinterlegtem Taragewicht durchgeführt.</p>	<p>Gewichte mit vorab hinterlegtem Tara-Wert verrechnen</p> <p><math>MEASUREMENT\_PROCESSING\_TARE = 100</math></p>
4.2.2.10	<p><u>Eine der in 4.2.2.0 genannten Bedingungen trifft zu:</u>  Über den Aufruf der Bibliotheksfunktion <i>Measurement_GetResult()</i> nacheinander durch Angabe von <i>MEASUREMENT_GET_GROSS</i> , <i>MEASUREMENT_GET_TARE</i> und <i>MEASUREMENT_GET_NET</i> als ersten Parameter, das Brutto-, Tara- und Nettogewicht abfragen.  Die Ergebnisse in den zugehörigen globalen Variablen <i>Weight_GrossWeight</i>, <i>Weight_TareWeight</i> und <i>Weight_NetWeight</i> speichern.</p>	<p>Gewichtswerte auslesen</p> <p><math>WEIGHT\_WEIGHTCHANNEL = 0</math>  <math>MEASUREMENT\_GET\_GROSS = 0</math>  <math>MEASUREMENT\_GET\_TARE = 1</math>  <math>MEASUREMENT\_GET\_NET = 2</math></p>



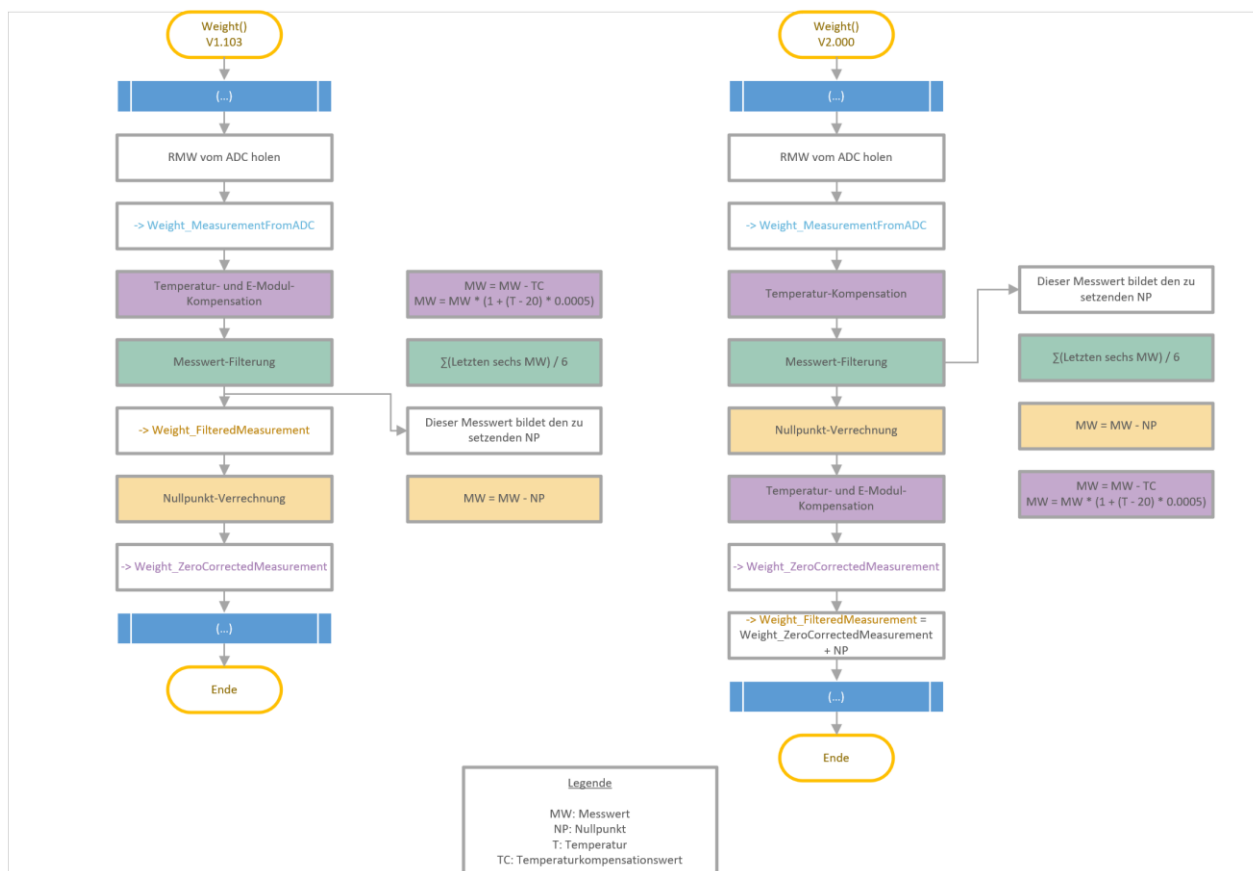
4.2.2.11	<p><u>Eine der in 4.2.2.0 genannten Bedingungen trifft zu:</u></p> <p>Zur Ermittlung des kompensierten Netto-Rohmesswerts ist zunächst der Kalibrierfaktor des Gewichtskanals auszulesen. Dazu die Bibliotheksfunktion <i>Measurement_GetCalibrationFactor()</i> aufrufen. Anschließend das Taragewicht <i>Weight_TareWeight</i> dividiert durch den Kalibrierfaktor vom kompensierten Rohmesswert <i>Weight_ZeroCorrectedMeasurement</i> abziehen. Das Ergebnis ist in der Globalen <i>Weight_ZeroCorrectedMeasurementWithTare</i> abzuliegen.</p>	<p>Kompensierten Netto-Rohmesswert berechnen</p> <p><i>WEIGHT_WEIGHTCHANNEL = 0</i></p>
4.2.2.12	<p><u>Alle 100ms Überprüfung auf Gewichtsbewegung</u></p> <p>100ms-Zeitgeberflag <i>Flag100ms</i> auslesen Ist dies gesetzt, für die weitere Verarbeitung die Motionparameter mittels der Funktion <i>Weight_GetMotionParameter()</i> auslesen.</p>	<p>100ms-Zeitgeberflag prüfen</p>
4.2.2.13	<p><u>100ms-Zeitgeberflag ist gesetzt:</u></p> <p>Den Zähler <i>g_Weight_uExecuteMotionCounter</i> inkrementieren und mit dem Motion-Filter <i>MotionParameter.byMotionFilter</i> vergleichen, um den Ablauf des Zeitfensters zu prüfen.</p>	<p>Motion-Zeitfenster prüfen</p>
4.2.2.14	<p><u>Zeitfenster abgelaufen</u> <u>(<i>g_Weight_uExecuteMotionCounter++ &gt;= MotionParameter.byMotionFilter</i>):</u></p> <p><i>g_Weight_uExecuteMotionCounter</i> nullen und über die Bibliotheksfunktion <i>Measurement_Processing()</i> mit <i>MEASUREMENT_PROCESSING_MOTION</i> als ersten Parameter, die Überprüfung auf Gewichtsbewegung ausführen.</p>	<p>Überprüfung der Gewichtsbewegung</p> <p><i>MEASUREMENT_PROCESSING_MOTION = 5</i></p>

### 3 Flowchart



## 4 Gegenüberstellung - Weight() - V1.103 und V2.000

Nachfolgende Übersicht soll die Veränderung der Ausgangsversion zum aktuellen Update zeigen



## **5 Kommentare**

## **6 Anhang**