



MOBA Mobile Automation AG

# Spezifikation

## *Main\_UpdateRetains()*

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	Main_UpdateRetains() .....	6
2.1	Beschreibung .....	6
2.2	Spezifikation.....	7
3	Kommentare.....	11
4	Anhang.....	12

# 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

*Main\_UpdateRetains()* der Datei *Main.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 Main\_UpdateRetains()**

### **2.1 Beschreibung**

Über die Dauer der Firmwareentwicklung, wurden immer wieder zusätzliche Variablen im Eepromspeicher abgelegt. Um eine Kompatibilität dieses Updates mit älteren Firmware-Versionen zu gewährleisten, bedarf es der Neuanlegung der neu im Eeprom hinzugekommenen Variablen und deren Initialisierung.

Im Bereich des Statistik-Moduls kam es im Laufe der Zeit zu einer Verschiebung zweier Einträge im Eeprom. Aus diesem Grund wird der komplette Eeprombereich dieses Moduls neu initialisiert und die Statistik gelöscht.

## 2.2 Spezifikation

Alle Spezifikationen sind in aufsteigender Reihenfolge zu erfüllen!

Weight()		
Index	Parameter	Datentyp
5.2.0.0	./.	void
Rückgabe		Datentyp
5.2.1.0	<u>Status der Funktionsausführung</u> <ul style="list-style-type: none"> <li>0: Fehlerfreie Ausführung</li> <li>&lt;&gt;0: Es ist ein Fehler bei der Ausführung aufgetreten (ggf. kann über Flags angezeigt werden, wo der Fehler aufgetreten ist)</li> </ul>	unsigned long
Verhalten		Bemerkung
5.2.2.0	Die Reorganisation des Eeproms ist nur einmalig zulässig, sofern der Durchlauf erfolgreich abgeschlossen wurde. Im Fehlerfall schaltet das System in den Sicherheitszustand.	LOAD_SAVE_VERSION = 212
	Die im weiteren Verlauf genannten Daten sind im Eeprom mit Standardwerten anzulegen. Statistikdaten werden dabei gelöscht und neu initialisiert.	
	Nach der Reorganisation muss an Adresse 0x418 (1048 <sub>dez.</sub> ) des Eeproms der Eintrag 0x000007D0 (2000 <sub>dez.</sub> ) liegen (Firmware-Version).	
5.2.2.1	Stimmen die Angaben zur Version im Code- und Eepromspeicher nicht überein, ist die Reorganisation durchzuführen.	Notwendigkeit der Eeprom-Reorganisation?  LOAD_SAVE_VERSION = 212
	Ist keine Notwendigkeit der Eeprom-Reorganisation erkannt worden, wird die Funktion mit der Rückgabe von 0 verlassen.	
5.2.2.2	<u>Reorganisation durchführen:</u> Dieser Prozess ist in mehreren Schritten durchzuführen. Jeder einzelne wird nur dann durchgeführt, wenn kein Fehler bis dato aufgetreten ist. Nach dem Setzen des letzten Parameters, wird die Firmware-Version als Longwert im Eeprom eingetragen - Adresse 0x418 (1048 <sub>dez.</sub> ). Ferner muss ein abschließender Software-Reset erfolgen. Dies stellt sicher, dass alle Parameter erneut aus dem Eeprom ausgelesen werden. Hierzu die Funktion <i>System-Reset()</i> aufrufen. Um zu gewährleisten, dass der Hardware-Watchdog zu diesem Zeitpunkt auch freigegeben ist, die Watchdog-Initialisierung vor dem Reset aufrufen ( <i>Watchdog_Ini()</i> ).	
5.2.2.3	<u>Reorganisation durchführen – Schritt 1 (RS232-Baudrate):</u> Es ist die RS232-Baudrate im Eeprom mit 9600 (baud) an Adresse 0x110 (272 <sub>dez.</sub> ) einzutragen. Dazu bedient man sich der Funktion <i>Save_Parameter(LOAD_SAVE_RS232_BAUDRA</i>	Default RS232-Baudrate einstellen (9600baud)

	<p><i>TE,...,4).</i></p> <p><i>Save_Parameter()</i> liefert im Fehlerfall einen Wert ungleich 0 zurück.</p> <p>In diesem Fall muss der Rückgabewert der Funktion einem Wert ungleich 0 entsprechen.</p>	LOAD_SAVE_RS232_BAUDRATE
5.2.2.4	<p><u>Reorganisation durchführen – Schritt 1 (ADC-Wandlungsrate):</u></p> <p>Bedingung zur Ausführung dieses Schritts ist, dass bisher kein Fehler aufgetreten ist.</p> <p>Es ist die ADC-Wandlungsrate im Eeprom mit <i>SYSTEM_CND_DEFAULT_CONVERSION_RATE_WEIGHT</i> (5.35) an Adresse 0x16 (22<sub>dez.</sub>) einzutragen. Dazu bedient man sich der Funktion <i>Save_Parameter(LOAD_SAVE_MEASUREMENT_CONVERSIONRATE,...,4).</i></p> <p><i>Save_Parameter()</i> liefert im Fehlerfall einen Wert ungleich 0 zurück.</p> <p>In diesem Fall muss der Rückgabewert der Funktion einem Wert ungleich 0 entsprechen.</p>	<p>Default ADC-Wandlerrate einstellen (5.35Hz)</p> <p><i>SYSTEM_CND_DEFAULT_CONVERSION_RATE_WEIGHT</i> = 5.35</p> <p><i>LOAD_SAVE_MEASUREMENT_CONVERSIONRATE</i> = 2</p>
5.2.2.5	<p><u>Reorganisation durchführen – Schritt 2 (SPI-Fehlerzähler):</u></p> <p>Bedingung zur Ausführung dieses Schritts ist, dass bisher kein Fehler aufgetreten ist.</p> <p>Es ist der SPI-Fehlerzähler im Eeprom mit 0 an Adresse 0x10E (270<sub>dez.</sub>) einzutragen. Dazu bedient man sich der Funktion <i>Save_Parameter(LOAD_SAVE_SPI_FAULT_COUNTER,...,1).</i></p> <p><i>Save_Parameter()</i> liefert im Fehlerfall einen Wert ungleich 0 zurück.</p> <p>In diesem Fall muss der Rückgabewert der Funktion einem Wert ungleich 0 entsprechen.</p>	<p>SPI-Fehlerzähler nullen</p> <p><i>LOAD_SAVE_SPI_FAULT_COUNTER</i> = 100</p>
5.2.2.6	<p><u>Reorganisation durchführen – Schritt 3 (Kein Reset bei mehrfachen SPI-Fehlern):</u></p> <p>Bedingung zur Ausführung dieses Schritts ist, dass bisher kein Fehler aufgetreten ist.</p> <p>Es ist der Flag ‚Kein Reset bei SPI-Fehler‘ im Eeprom mit 1 an Adresse 0x10F (271<sub>dez.</sub>) einzutragen. Dazu bedient man sich der Funktion <i>Save_Parameter(LOAD_SAVE_FLAG_DONOT_RESET,...,1).</i></p> <p>Diese gibt im Fehlerfall einen Wert ungleich 0 zurück.</p> <p><i>Save_Parameter()</i> liefert im Fehlerfall einen Wert ungleich 0 zurück.</p> <p>In diesem Fall muss der Rückgabewert der Funktion einem Wert ungleich 0 entsprechen.</p>	<p>Flag 'Kein Reset bei mehrfachen SPI-Fehlern' setzen</p> <p><i>LOAD_SAVE_FLAG_DONOT_RESET</i> = 101</p>
5.2.2.7	<p><u>Reorganisation durchführen – Schritt 4 (RS232-Kommunikationssperre):</u></p> <p>Bedingung zur Ausführung dieses Schritts ist, dass bisher kein Fehler aufgetreten ist.</p> <p>Es ist der Code zur RS232-Kommunikationssperre im Eeprom mit 0 an Adresse 0x114 (276<sub>dez.</sub>) einzutragen. Dazu bedient man sich der Funktion <i>Save_Parameter(LOAD_SAVE_RS232_DISABLE_CODE,...,4).</i></p>	<p>Code zur RS232-Kommunikationssperre löschen</p>



	<p><i>Save_Parameter()</i> liefert im Fehlerfall einen Wert ungleich 0 zurück.</p> <p>In diesem Fall muss der Rückgabewert der Funktion einem Wert ungleich 0 entsprechen.</p>	<p>LOAD_SAVE_RS232_DISABLE_CODE = 103</p>
5.2.2.8	<p><u>Reorganisation durchführen – Schritt 5 (Proportionalanteil der Stromeinstellung):</u></p> <p>Bedingung zur Ausführung dieses Schritts ist, dass bisher kein Fehler aufgetreten ist.</p> <p>Es ist der Proportionalanteil der Stromeinstellung im Eeprom mit 50 an Adresse 0x1B4 (436<sub>dez.</sub>) einzutragen. Dazu bedient man sich der Funktion <i>Save_Parameter(LOAD_SAVE_PROPORTIONALPORTION_FEEDBACK,...,1)</i>.</p> <p><i>Save_Parameter()</i> liefert im Fehlerfall einen Wert ungleich 0 zurück.</p> <p>In diesem Fall muss der Rückgabewert der Funktion einem Wert ungleich 0 entsprechen.</p>	<p>Proportionalanteil der SetDAC-Routine auf 50% setzen</p> <p>LOAD_SAVE_PROPORTIONALPORTION_FEEDBACK = 60</p>
5.2.2.9	<p><u>Reorganisation durchführen – Schritt 6 (Neuinitialisierung der Statistikdaten):</u></p> <p>Bedingung zur Ausführung dieses Schritts ist, dass bisher kein Fehler aufgetreten ist.</p> <p>Über die Funktion <i>Statistics_Ini(1)</i> wird sowohl der Relativ- als auch der Absolutstatistikspeicher neu initialisiert.</p> <p>Liegt zu diesem Zeitpunkt ein Fehler an, muss der Rückgabewert der Funktion einem Wert ungleich 0 entsprechen.</p>	<p>Gesamte Statistik auf Defaultwerte setzen - Relativ- und Absolutstatistik</p>
5.2.2.10	<p><u>Reorganisation durchführen – Schritt 6 (Eintrag der Firmware-version):</u></p> <p>Bedingung zur Ausführung dieses Schritts ist, dass bisher kein Fehler aufgetreten ist.</p> <p>Es ist die Firmwareversion im Eeprom an Adresse 0x418 (1048<sub>dez.</sub>) einzutragen. Dazu bedient man sich der Funktion <i>Save_Parameter(LOAD_SAVE_VERSION,..., 4)</i></p> <p><i>Save_Parameter()</i> liefert im Fehlerfall einen Wert ungleich 0 zurück.</p> <p>In diesem Fall muss der Rückgabewert der Funktion einem Wert ungleich 0 entsprechen.</p> <p>War die Funktionsausführung bis hierher fehlerfrei, ist ein Reset auszuführen. Dies sorgt für ein erneutes Einlesen der nun neu angelegten Daten im Eeprom.</p> <ul style="list-style-type: none"> <li>• Watchdog initialisieren – <i>Watchdog_Ini()</i></li> <li>• Danach Reset ausführen – <i>System_Reset()</i></li> </ul>	<p>Aktuelle Firmwareversion im Eeprom eintragen</p> <p>LOAD_SAVE_VERSION = 212</p>
5.2.2.11	<p>Liegt zu diesem Zeitpunkt ein Fehler an oder der System-Reset wird übersprungen, ist das System in den Sicherheitszustand zu setzen. Dies geschieht über die Funktion <i>System_SetSystemState(SYSTEM_ERROR)</i></p>	<p>Überführung in den Sicherheitszustand bei Fehler</p> <p>SYSTEM_ERROR = 255</p>
5.2.2.12	<p>Die Funktion gibt den Status der Funktionsausführung zurück (u32RetVal)</p>	



### 3 Kommentare

[illegible]

**Speicherauszug nach Eeprom-Update  
(hier noch mit Version V1.200 @0x0418-0x041B)**

## 4 Anhang