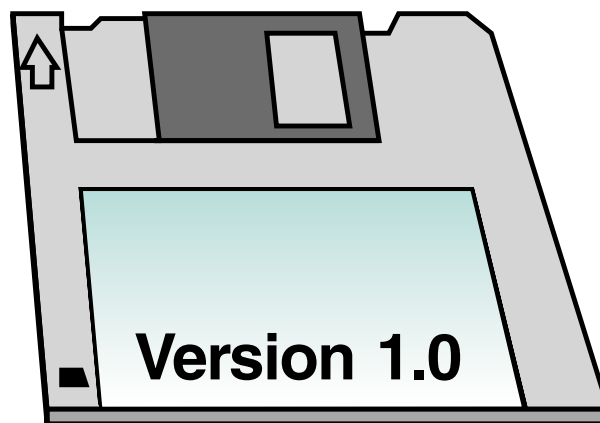


COMBICOM



KEB Parametrierkanal-Treiber für Siemens S7
KEB Parameterizing Channel-Driver for Siemens S7

(D)

Seite D3 D21

(GB)

Page GB3 GB21

Inhaltsverzeichnis

1.	Allgemeines	4
2.	Benötigte Unterlagen, Literaturverzeichnis, Inhalt der Diskette	4
3.	Einleitung	5
3.1	Was ist der Parametrierkanal	5
4.	Grundsätzliche Funktionsweise des Parametrierkanal-Treibers	6
5.	Vorbereitung für die Anwendung des Treibers ..	7
5.1	Aufbau der Testumgebung	7
5.2	Installation des Treibers	8
6.	Die Anwenderschnittstelle	11
6.1	Zusammenspiel von Anwendungsprogramm und Parametrier-Treiber	14
6.1.1	Konfiguration der Treiber-Software	14
6.1.2	Zyklischer Betrieb der Anwendungs-Software	14
6.2	Aufbau des Slave-DB	15
6.3	Aufbau des Slave-Listen-DB	17
6.4	Aufbau des Init-DB	18
6.5	Timeout-Überwachung	19
6.6	PROFIBUS-Diagnose	19
7.	Randbemerkungen	20
7.1	Konfiguration von KEB Slaves	20
7.2	Zugriff auf konsistente Peripheriebereiche bei SIMATIC S7	20
8.	Belegte Programm-Ressourcen des Treibers ...	21

1. Allgemeines

Die vorliegenden Unterlagen sowie die Hard- und Software sind Entwicklungen der Karl E. Brinkmann GmbH. Irrtum vorbehalten. Die Karl E. Brinkmann GmbH hat diese Unterlagen, die Hard- und Software nach bestem Wissen erstellt, übernimmt aber nicht die Gewähr dafür, daß die Spezifikationen den vom Anwender angestrebten Nutzen erbringen. Die Karl E. Brinkmann GmbH behält sich das Recht vor, Spezifikationen ohne vorherige Ankündigung zu ändern oder Dritte davon in Kenntnis zu setzen.

2. Benötigte Unterlagen, Literaturverzeichnis, Inhalt der Diskette

Zum Durchführen des Tests benötigen Sie folgende Dokumente:

- [1]: Diese Anwenderbeschreibung.
- [2]: Bedienungsanleitung der eingesetzten KEB PROFIBUS Anschaltung.
- [3]: Applikationsanleitung des eingesetzten KEB Frequenzumrichters/Servos.

S7_KEB1N.ARJ Der Inhalt der Treiber-SW-Diskette:
Archiv des Beispiel-Projektes für Simatic-Manager inklusive:

- Systemdaten (Hardware-Konfiguration)
- alle Bausteine (FCs, DBs, OBs)
- Symboltabelle
- Variablenliste VAT1 für Variablen Steuer/Beobachten
- Diese Anwenderbeschreibung

KEB_DP1.GSD GSD-Datei für „**KEB-Gateway DP**“:
Ist sowohl für das KEB PROFIBUS-DP-Gateway als auch für den F4/S4-PROFIBUS-DP-Operator zu verwenden.

3. Einleitung

KEB-Frequenzumrichter und Servos können optional mit einer PROFIBUS-DP-Schnittstelle ausgerüstet werden. Der PROFIBUS-DP ist ein prozessgesteuerter Feldbus für dezentrale Peripherie. Standardisiert ist der PROFIBUS-DP in der europäischen Norm **EN50170**.

In der Automatisierungstechnik ist die Siemens Simatic SPS vor allem im deutschsprachigen Raum sehr weit verbreitet. Dabei ist die Umstellung von der Simatic S5- auf die Simatic S7-Reihe immer deutlicher spürbar. Um einen großen Anwenderkreis bei der Realisierung seiner Automatisierungsaufgaben zu unterstützen, entwickelte KEB eine S7-Treiber-Software. Es handelt sich hier um ein Programmpaket, das den sog. **Parametrierkanal** eines KEB PROFIBUS-DP-Gerätes bedient. Der Parametrierkanal beschreibt die Funktionalität eines jeden KEB DP-Slaves, über den der Anwender auf jeden beliebigen Parameter des KEB Gerätes zugreifen kann. Alternativ kann der Anwender auf einige vorher festgelegte Parameter lesend und schreibend über die sog. **Prozessdaten** zugreifen. Der Zugriff auf die Prozessdaten ist sehr einfach und bedarf deshalb auch keines SW-Treibers. Aus diesem Grund beschränkt sich die im folgenden beschriebene Software auch ausschließlich auf den Parametrierkanal.

Um den Umfang dieser Beschreibung möglichst knapp zu halten, werden hier keine detaillierten Erläuterungen zum Ablauf des Parametrierkanalprotokolls aufgeführt. Das Ziel dieses Treibers ist, den Anwender von der Notwendigkeit dieses Detailwissens zu befreien. Jedoch sind selbstverständlich alle Einzelheiten zum Parametrierkanal sowie allen anderen Spezifikationen eines KEB PROFIBUS-DP-Gerätes in der Bedienungsanleitung der eingesetzten KEB PROFIBUS-Anschaltung nachzulesen.

3.1 Was ist der Parametrierkanal

Wie oben schon kurz erwähnt, handelt es sich bei dem Parametrierkanal um einen Teil der Funktionalität eines KEB PROFIBUS-DP-Gerätes. Beim KEB PROFIBUS-DP werden nach einer kurzen Initialisierungsphase zyklisch Nutzdaten zwischen einem Master und den ihm zugeteilten Slave ausgetauscht. Ein Teil dieser Nutzdaten, nämlich immer die ersten 8 Byte, beinhalten **bei KEB Slaves** den Parametrierkanal. Über diese 8 Byte kann der Master den Wert eines zu adressierenden Parameters erfragen (Lesen) oder den Wert ändern (Schreiben). Dabei geschieht die Abwicklung eines Parametrierkanal-Auftrages über mindestens zwei Datenzyklen zwischen Master und Slave. Desweiteren ist anzumerken, daß jeder Parametrierkanal-Dienst bestätigt ist. D.h. Der Master fordert das Lesen oder Schreiben eines bestimmten Parameters beim Slave an und bekommt darauf eine Bestätigung. So ist es möglich, zu erkennen, ob der Slave den Dienst wirklich ausgeführt hat.

Vollständig beschrieben wird ein Parametrierkanal-Dienst durch seine im folgenden aufgeführten Parameter:

<i>Index</i>	Zusammen mit Subindex gibt der Index die Parameter-Adressierung vor.
<i>Subindex</i>	Zusammen mit Index gibt der Subindex die Parameter-Adressierung vor.
<i>Dlen</i>	Die Datenlänge des adressierten Parameters in Byte.
<i>Daten</i>	Der Wert des adressierten Parameters.
<i>Dienst-Code</i>	Bestimmt, ob ein Schreib- oder ein Lesedienst ausgeführt werden soll.
<i>Fehler-Code</i>	Zeigt an, ob bei der Ausführung eines Dienstes ein Fehler aufgetreten ist.

Durch die Beschränkung des Parametrierkanals auf 8 Byte können über ihn lediglich Parameter mit einer Datenlänge von maximal 4 Byte verändert oder gelesen werden. Dies bedeutet keinerlei Beschränkung, da alle Parameter von KEB Geräten der F4/S4-Reihe und früherer Generationen eine Datenbreite von 2 Byte besitzen. Alle Konfigurationsparameter der KEB PROFIBUS-DP-Anschaltung und auch die KEB DRIVECOM-Profil-Parameter sind mit dieser Einschränkung ebenfalls nutzbar.

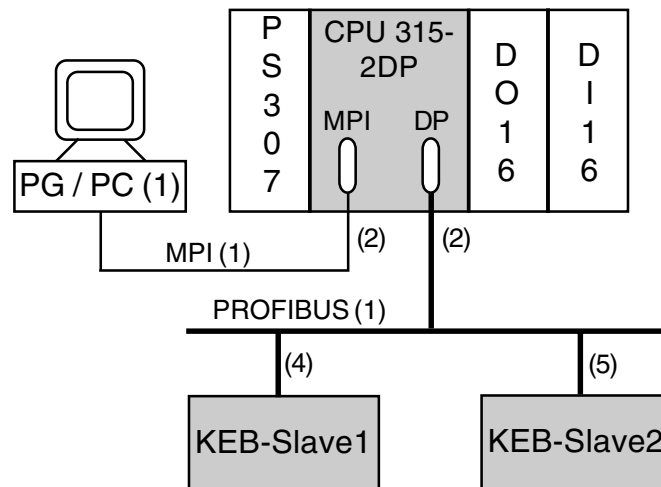
4. Grundsätzliche Funktionsweise des Parametrierkanal-Treibers

Der hier realisierte S7-Treiber besteht im Prinzip aus zwei Teilen. Der erste Teil stellt den eigentlichen Parametrierkanaltreiber dar. Er besteht im wesentlichen aus einer zyklisch aufzurufenden Funktion (FC2, DO_PARA) und für jeden KEB PROFIBUS-Slave aus einem Datenbaustein, über den jeweils ein Dienst abgewickelt wird.

Darauf aufbauend stellt die hier beschriebene SW aber noch eine minimale Anwendungsfunktion bereit. Diese besteht darin, daß in einem Verwaltungs-Datenbaustein (SL_LIST) eine Slave-Liste eingetragen wird. In dieser Liste gibt es für jeden KEB-Slave einen Eintrag „Slave-DB-Nr“ und einen Eintrag für eine „Initialisierungs-DB-Nr“. Der zweite Teil des Treibers stellt nun sicher, daß zyklisch alle in der Liste eingetragenen Slaves bearbeitet werden. Desweiteren werden nach dem SPS-Neustart einmalig zu jedem Slave der eine Init-DB-Nr eingetragen hat (Init_DBNR != 255), die Aufträge, die der Init-DB enthält zu diesem Slave geschrieben.

Im zyklischen Betrieb, d.h. wenn alle Slaves initialisiert sind, kann dann das Anwendungsprogramm für jeden Slave in dessen Slave-DB Aufträge eintragen. Durch die zyklische Bearbeitung wird dieser dann abgearbeitet und die Bestätigung im Slave-DB eingetragen sobald sie vorhanden ist. Der Parametrierkanal-Treiber enthält auch eine Timeout-Überwachung. D.h., daß ein Parametrierkanal-Auftrag vom Treiber mit einem Fehlercode beendet wird, wenn innerhalb der Timeoutzeit vom Slave keine Bestätigung empfangen wurde (s.u.).

5. Vorbereitungen für die Anwendung des Treibers



5.1 Aufbau der Testumgebung

Um den Test des Treibers durchführen zu können, ist der o.a. Testaufbau notwendig. Generell sind zwei Möglichkeiten der PROFIBUS-Ankopplung von KEB Geräten möglich. Für beide Möglichkeiten wird die notwendige Konfiguration im folgenden kurz zusammengefaßt:

PROFIBUS-Ankopplung über PROFIBUS-DP-Operator (s. [2]):

In diesem Fall besteht ein KEB Slave aus einem KEB Frequenzumrichter/Servo plus aufgestecktem PROFIBUS-DP-Operator.

- Parametrierkanal immer aktiviert.
- Einstellung der PROFIBUS-Adresse(Ts) über Vorgabe der Umrichteradresse (ud.06), Ts = ud.06 .
- Keine Einstellung der PROFIBUS-Baudrate notwendig, da standardmäßig mit automatischer Bitratenerkennung gearbeitet wird.

PROFIBUS-Ankopplung über PROFIBUS-DP-Gateway (s. [2]):

In diesem Fall besteht ein KEB Slave aus einem KEB Frequenzumrichter/Servo, plus Interface-Operator, plus ein extern über die serielle Schnittstelle angeschlossenes PROFIBUS-DP-Gateway. Um eine reibungslose Funktion zu gewährleisten, sind die Parameter Umrichteradresse und KEB DIN66019-Baudrate von Frequenzumrichter und Gateway aufeinander abzustimmen. Standardmäßig sind im Gateway als S_addr = 1 und KEB_DIN66019_Baud = 3(9600 Bit/s) eingestellt. Die gleichen Werte muß der angeschlossene Frequenzumrichter/Servo enthalten (ud.06=1, ud.07=3). Stellen Sie bitte zusätzlich sicher, daß in den angeschlossenen Frequenzumrichtern/Servos das Buspasswort (ud.01) = Application (440) folgendes beinhaltet:

- Parametrierkanal, aktiviert durch Stellung des Schalters S2-1 auf ON.
- Einstellung der PROFIBUS-Adresse (Ts) über Einstellung des 8poligen DIL-Schalters S1 am Gateway.
- Einstellung der PROFIBUS-Übertragungsgeschwindigkeit über Einstellung des 4poligen DIL-Schalters S2 (S2-2...S2-4) am Gateway.

Wenn nur ein KEB Gerät mit PROFIBUS-DP-Anschaltung vorliegt, muß an der Hardware-Konfiguration des Treibers zunächst nichts geändert werden. In diesem Fall wird jedoch die rote LED **BUSF** an der CPU 315-2DP zyklisch blinken, da nicht alle projektierten Slaves vorhanden sind. Dennoch kann der Test mit einem KEB Slave durchgeführt werden.

5.2 Installation des Treibers

Bei der Erstellung des Treibers wurde mit dem **Simatic Manager STEP7 S7/M7 Version 4.02.1** gearbeitet. Entwickelt wurde auf Bausteinebene. Es gibt aus diesem Grund keine **Quellen**. Der Treiber besteht aus einem kompletten Projekt, in dem alle notwendigen Daten vorhanden sind. Bevor der Anwender beginnen kann, muß das Projekt zunächst einmal in der Anwender-SPS installiert werden. Ausgeliefert wird das Projekt als Archiv, das Sie im Simatic-Manager zunächst **Dearchivieren** müssen. Dies geschieht im Simatic Manager unter dem Menue **Datei->Dearchivieren**.

In dem Projekt ist folgende Hardware-Konfiguration voreingestellt:

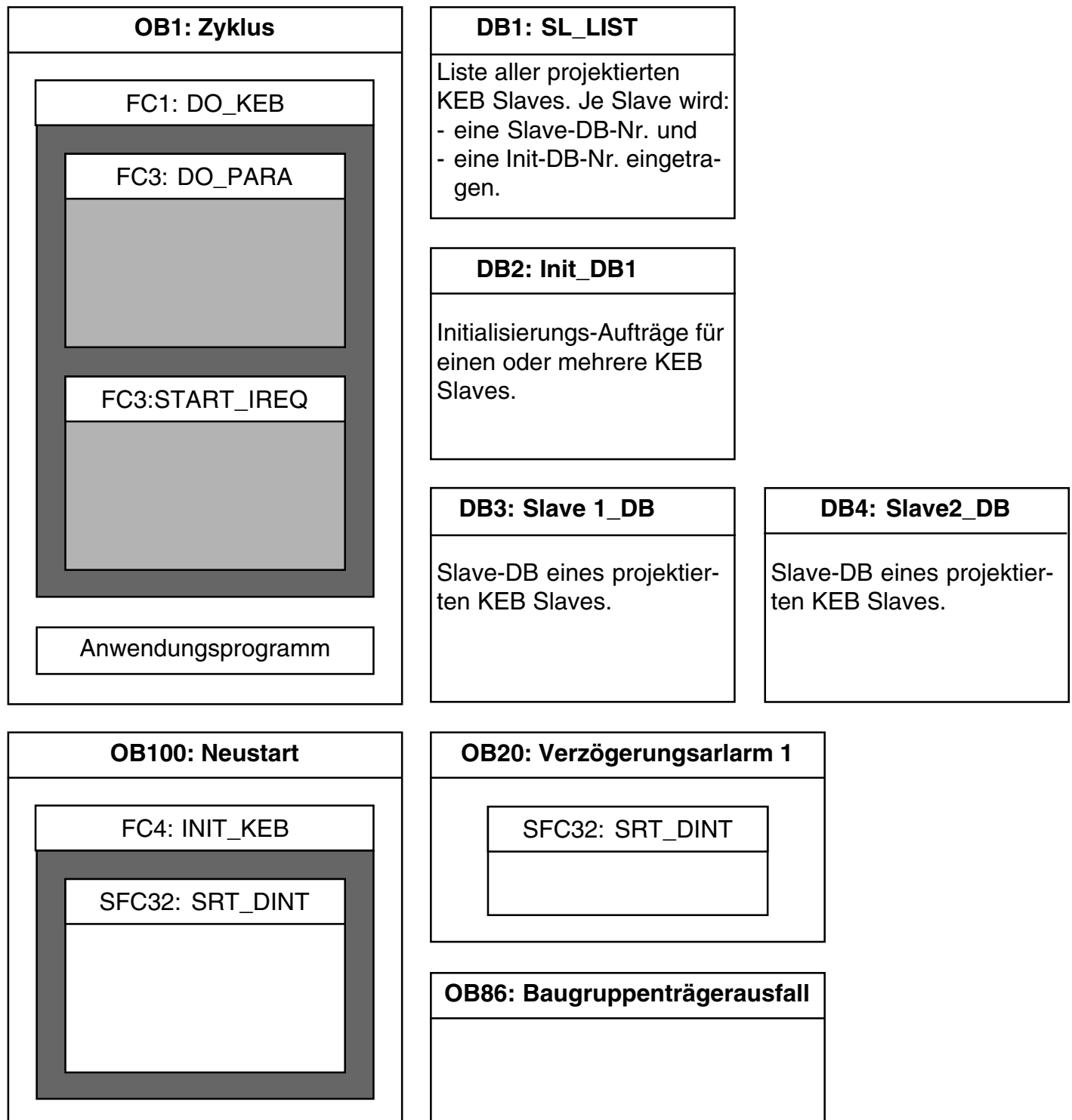
- 1 Simatic 300-Station, bestehend aus:
 - 1 Laststromversorgung PS307 5A in Steckplatz 1,
 - 1 CPU315-2DP als PROFIBUS-DP-Master(Ts=2) am Subnetz PROFIBUS(1) in Steckplatz 2. Die CPU ist weiterhin als MPI-Teilnehmer verbunden mit PG/PC(1), MPI-Adresse = 2
 - 1 digitale Ausgabebaugruppe DO16*DC24V/0.5A an Steckplatz 4
 - 1 digitale Eingabebaugruppe DI16*DC24V an Steckplatz 5
- 2 KEB Gateway DP als PROFIBUS-DP-Slaves(Ts=4,5) an Subnetz PROFIBUS(1).
- 1 PG/PC(1) als MPI-Teilnehmer an Subnetz MPI(1), MPI-Adresse = 0

Eventuell muß die o.g. Hardware-Konfiguration an Ihre Bedürfnisse angepaßt werden. Für einen ersten Test der SW ist es von Vorteil, wenn die Standard-Hardware-Konfiguration unverändert bleiben könnte. Die Hardware-Konfiguration muß dann in die SPS geladen werden.

Anschließend muß das S7-Programm in die SPS geladen werden. Hierzu werden z. B. in der Ansicht der Bausteine alle OBs, FCs und DBs in die SPS geladen. Die aufgeführten SFC's müssen in der SPS bereits vorhanden sein.

Die Bausteine des Treibers sind im einzelnen:

OB1, Zyklus	Enthält das zyklische Hauptprogramm der kompletten Anwendung. Besteht bei Auslieferung lediglich aus dem Aufruf der Funktion FC1 (DO_KEB).
OB100, Neustart	Enthält nach Auslieferung lediglich den Aufruf der Funktion FC4 (INIT_KEB).
OB20, Verzögerungsalarm 1	Mit dem Verzögerungsalarm wird ein zyklischer Alarm realisiert, der mit einer Zykluszeit von 100ms aufgerufen wird. Genutzt wird er für die Timeouterkennung des Parametrierkanal-Treibers.
FC1, DO_KEB	Anwendungsteil des Treibers. Er stellt sicher, daß zu Beginn alle konfigurierten KEB Slaves mit einem Init_DB entsprechend initialisiert werden. Nach der Initialisierung sorgt der FC1 dafür, daß alle Slaves vom Parametrierkanal-Treiber bearbeitet werden.
FC2, DO_PARA	Wird von FC1 aufgerufen. Bearbeitet die Parametrieraufträge vollständig.
FC3, START_IREQ	Wird vom FC1 aufgerufen und startet einen neuen Initialisierungsauftrag.
FC4, INIT_KEB	Übernimmt die komplette Initialisierung des Treibers (s. OB100).
DB1, SL_LIST	Verwaltungsbaustein für den Treiber, enthält neben Verwaltungsinformationen eine Slave-Liste mit je einem Eintrag; Slave-DB-Nr und Init-DB-Nr je Slave.
DB2, INIT1_DB	Enthält beispielhaft eine Liste von Initialisierungsaufträgen, die zu beiden KEB Slaves während der Initialisierungsphase geschrieben werden. Dabei handelt es sich um Aufträge, die die Standard-Prozessdatenbelegung schreibt.
DB3, Slave1_DB	Slave-Datenbaustein des 1. KEB Slave.
DB4, Slave2_DB	Slave-Datenbaustein des 2. KEB Slave.



6. Die Anwenderschnittstelle

In diesem Kapitel erfahren Sie, was Sie tun müssen, um den Parametrierkanal-Treiber in Ihrem Anwendungsprogramm zu nutzen. Dabei soll zunächst der erste Versuch mittels **Variable Beobachten/Steuern** vom PG/PC aus durchgeführt werden. An dieser Stelle wird davon ausgegangen, daß die SW komplett in der SPS geladen und die CPU im Zustand STOP ist. Ebenso wird vorausgesetzt, daß die generelle PROFIBUS-Kommunikation läuft. Dies ist daran zu erkennen, daß die beiden roten LED's **SF DP** und **BUSF** an der CPU nicht leuchten bzw. blinken.

In der Ansicht der Bausteine kommen Sie durch Doppelklick auf **VAT1** direkt in das Menue Variable Beobachten/Steuern. Die dort aufgeführten Variablen zeigen zum einen den aktuellen Bearbeitungsstand des Treibers an (DB1), zum anderen werden die wichtigsten Variablen der beiden Slave-Datenbausteine (DB3, DB4) angezeigt.

Achten Sie darauf, daß im Menue **Variable->Trigger** die Triggerbedingungen wie folgt eingestellt sind:

- Triggerpunkt für Beobachten: Zyklusbeginn,
Triggerbedingung für Beobachten: permanent.
- Triggerpunkt für Steuern: Zyklusbeginn,
Triggerbedingung für Steuern: einmalig.

Aktivieren Sie das Variablen Beobachten.

Wenn Sie nun die CPU in **RUN-P** schalten, wird die CPU anlaufen und die Initialisierung der Slaves wird durchgeführt. Abgeschlossen ist die Initialisierung der Slaves, wenn die Variable DB1.DBB0 den Wert 3 enthält. Nun können Sie über das Steuern der Variablen-Werte von DB3, DB4 Aufträge an den 1. oder 2. Slave oder auch an beide gleichzeitig absenden.

Beispiel In dem Beispiel können Sie versuchen, aus beiden Slaves den Wert des Parameters mit dem Index= 4606h und dem Subindex = 0 auszulesen. In den KEB Geräten der Baureihen F4 oder S4 befindet sich an dieser Adresse der Parameter mit dem Namen **Umrichteradresse**. Bei anderen Gerätetypen kann es notwendig sein, den Index entsprechend zu ändern, um einen anderen Parameterwert auszulesen. Siehe dazu die Applikationsanleitung des eingesetzten KEB Gerätes. Beachten Sie bitte, daß sich der **Index aus der Parameter-Adresse + 2000h** Offset ergibt. Sie müssen also die aus der Applikationsanleitung bestimmte Parameter-Adresse mit 2000 (hex) addieren, um auf den entsprechenden Index zu kommen, der im Slave-DB eingetragen wird.

Beispiel 1 Lesen des Parameters mit Index = 4606 (hex) und Subindex = 0: (**ud.06, Umrichteradresse**): Machen Sie die folgenden Eintragungen jeweils in der **Steuerwert**-Spalte der Variablen:

Operand	Symbol	Steuerwert
DB3.DBB0	„Slave1_DB“.State	B#16#01
DB3.DBD2	„Slave1_DB“.Error	Keinen Steuerwert vorgeben
DB3.DBW8	„Slave1_DB“.Index	W#16#4606
DB3.DBB10	„Slave1_DB“.Subindex	B#16#00
DB3.DBB11	„Slave1_DB“.Dlen	Keinen Steuerwert vorgeben
DB3.DBD12	„Slave1_DB“.Data	Keinen Steuerwert vorgeben
DB4.DBB0	„Slave2_DB“.State	B#16#01
DB4.DBD2	„Slave2_DB“.Error	Keinen Steuerwert vorgeben
DB4.DBW8	„Slave2_DB“.Index	W#16#4606
DB4.DBB10	„Slave2_DB“.Subindex	B#16#00
DB4.DBB11	„Slave2_DB“.Dlen	Keinen Steuerwert vorgeben
DB4.DBD12	„Slave2_DB“.Data	Keinen Steuerwert vorgeben

Die so vorbereiteten Steuerwerte senden Sie durch **Variable->Steuern** an die CPU. Wenn die Statuswerte von „Slave1_DB“.State und „Slave2_DB“.State den Wert "0" aufweisen, sind beide Aufträge abgearbeitet. Jetzt können Sie in „Slave1_DB“.Error und „Slave2_DB“.Error den Fehler-Code auswerten. Diese sollten den Wert "0" enthalten. Wenn dies der Fall ist, kann der gelesene Wert in „Slave1_DB“.Data und „Slave2_DB“.Data abgelesen werden. Wieviele der Datenbytes im Data-Feld relevant sind, geben die Statuswerte der Variablen „Slave1_DB“.Dlen und „Slave2_DB“.Dlen. In unserem Beispiel sollten die Statuswerte folgende Werte besitzen:

Operand	Symbol	Statuswert
DB3.DBB0	„Slave1_DB“.State	B#16#00
DB3.DBD2	„Slave1_DB“.Error	DW#16#00000000
DB3.DBW8	„Slave1_DB“.Index	W#16#4606
DB3.DBB10	„Slave1_DB“.Subindex	B#16#00
DB3.DBB11	„Slave1_DB“.Dlen	B#16#02
DB3.DBD12	„Slave1_DB“.Data	DW#16#0004XXXX
DB4.DBB0	„Slave2_DB“.State	B#16#00
DB4.DBD2	„Slave2_DB“.Error	DW#16#00000000
DB4.DBW8	„Slave2_DB“.Index	W#16#4606
DB4.DBB10	„Slave2_DB“.Subindex	B#16#00
DB4.DBB11	„Slave2_DB“.Dlen	B#16#02
DB4.DBD12	„Slave2_DB“.Data	DW#16#0005XXXX

Beispiel 2 Schreiben des Parameters mit Index = 4100 (hex) und Subindex = 0: (**op.00,Sollwertquelle**): Machen Sie die folgenden Eintragungen jeweils in der **Steuerwert**-Spalte der Variablen:

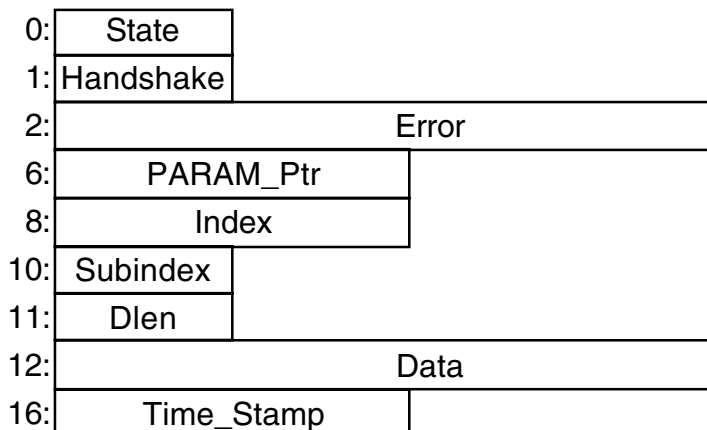
Operand	Symbol	Steuerwert
DB3.DBB0	„Slave1_DB“.State	B#16# 02
DB3.DBD2	„Slave1_DB“.Error	Keinen Steuerwert vorgeben
DB3.DBW8	„Slave1_DB“.Index	W#16# 4100
DB3.DBB10	„Slave1_DB“.Subindex	B#16# 00
DB3.DBB11	„Slave1_DB“.Dlen	B#16# 02
DB3.DBD12	„Slave1_DB“.Data	DW#16# 0001XXXX
DB4.DBB0	„Slave2_DB“.State	B#16# 02
DB4.DBD2	„Slave2_DB“.Error	Keinen Steuerwert vorgeben
DB4.DBW8	„Slave2_DB“.Index	W#16# 4100
DB4.DBB10	„Slave2_DB“.Subindex	B#16# 00
DB4.DBB11	„Slave2_DB“.Dlen	B#16# 02
DB4.DBD12	„Slave2_DB“.Data	DW#16# 1000XXXX

Die so vorbereiteten Steuerwerte senden Sie durch **Variable->Steuern** an die CPU. Wenn die Statuswerte von „Slave1_DB“.State und „Slave2_DB“.State den Wert "0" aufweisen, sind beide Aufträge abgearbeitet. Jetzt können Sie in „Slave1_DB“.Error und „Slave2_DB“.Error den Fehler-Code auswerten. In diesem Beispiel sollte „Slave1_DB“.Error den Wert "0" enthalten, „Slave2_DB“.Error jedoch den Wert DW#08000030. Dies bedeutet, daß Slave 2 den Schreibauftrag nicht durchgeführt hat, weil versucht wurde, einen ungültigen Wert vorzugeben (siehe Fehler-Codes in [2]).

Operand	Symbol	Statuswert
DB3.DBB0	„Slave1_DB“.State	B#16#00
DB3.DBD2	„Slave1_DB“.Error	DW#16# 00000000
DB3.DBW8	„Slave1_DB“.Index	W#16#4100
DB3.DBB10	„Slave1_DB“.Subindex	B#16#00
DB3.DBB11	„Slave1_DB“.Dlen	B#16#02
DB3.DBD12	„Slave1_DB“.Data	DW#16#0001XXXX
DB4.DBB0	„Slave2_DB“.State	B#16#00
DB4.DBD2	„Slave2_DB“.Error	DW#16# 08000030
DB4.DBW8	„Slave2_DB“.Index	W#16#4100
DB4.DBB10	„Slave2_DB“.Subindex	B#16#00
DB4.DBB11	„Slave2_DB“.Dlen	B#16#02
DB4.DBD12	„Slave2_DB“.Data	DW#16#1000XXXX

- 6.1 Zusammenspiel von Anwendungsprogramm und Parametrierkanal-Treiber**
- Zur Einbindung des Parametrierkanal-Treibers in Ihr Anwendungsprogramm ist folgendes Vorgehen empfehlenswert:
- Konfiguration der Treiber-Software.
 - Starten des Anwendungsprogramms.
 - Warten, bis „SL_LIST“.State (DB1.DBB0) = 3 ist.
 - Zyklischer Betrieb der Anwendungssoftware: Durchführen der gewünschten Parametrierkanal-Aufträge.
- 6.1.1 Konfiguration der Treiber-Software**
- Zunächst muß die Konfiguration des Treibers durchgeführt werden. Dazu muß der Slave-Listen-DB entsprechend der Anzahl am PROFIBUS angeschlossener KEB Slaves vorbereitet werden. Hierzu ist für jeden projektierten Slave eine Slave-DB-Nr (ungleich 255) einzutragen. Beachten Sie bitte, daß der Wert 255 als Slave-DB-Nr das Ende der Liste bedeutet. Alle eventuell nachfolgenden Slaves würden nicht mehr bearbeitet. Desweiteren ist für jeden zu initialisierenden Slave die gewünschte Init-DB-Nr (ungleich 255) einzutragen.
- Für jeden projektierten KEB Slave ist ein Slave-DB anzulegen mit einer Länge von 18 Byte (s.u.). In jedem dieser Slave-DB's ist der Parameter **Param_Ptr** mit der Peripherieadresse vorzubelegen, an dem der Parametrierkanal dieses Slaves beginnt. Nehmen Sie hierzu die Hardware-Konfiguration der einzelnen Slaves als Informationsquelle. Eventuell vorgesehene Initialisierungs-Datenbausteine sind nach dem Vorbild des DB2 entsprechend Ihren Wünschen zu generieren (s.u.).
- 6.1.2 Zyklischer Betrieb der Anwendungs-Software**
- Wenn die Initialisierungsphase des Parametrierkanal-Treibers abgeschlossen ist, können Sie Aufträge zu den einzelnen Slaves absetzen. Dabei kann je KEB Slave immer nur ein Auftrag gleichzeitig bearbeitet werden. Sie können aber durchaus gleichzeitig zu mehreren oder allen KEB Slaves jeweils einen Auftrag anschieben. Bei dem Durchführen eines Parametrierkanal-Auftrages gibt es folgende Schrittkette:
- Überprüfen Sie, ob der Parameter State in dem betreffenden Slave-DB = "0" ist. Nur dann darf ein neuer Auftrag angeschoben werden.
 - Tragen Sie den kompletten Auftrag im Slave-DB ein (State, Index, Subindex, Dlen (nur bei Schreiben), Data (nur bei Schreiben)), s.u.
 - Während Sie auf den Abschluß des Auftrags warten, stellen Sie sicher, daß der zyklische Aufruf des Parametrierkanal-Treibers (DO_KEB) gewährleistet ist. Denn nur so kann der Auftrag abgearbeitet werden. Der Auftrag ist abgeschlossen, wenn der Parameter State in dem Slave-DB auf den Wert "0" zurückgesetzt wurde.
 - Analysieren des Ergebnisses.

6.2 Aufbau des Slave-DB



byte State Bearbeitungsstatus: 0 = Idle (vom Treiber nach vollendeter Bearbeitung gesetzt)

1 = Lesen angefordert (vom Anwendungsprogramm gesetzt)

2 = Schreiben angefordert (vom Anwendungsprogramm gesetzt)

3 = Lesen läuft (vom Treiber gesetzt)

4 = Schreiben läuft (vom Treiber gesetzt)

dword Error Ergebnis:

0 = Kein Fehler aufgetreten

FFFFXXXXh: Fehlermeldung vom S7-Betriebssystem

mit XXXXh = Fehlercode der Betriebssystemfunktionen, s. Beschreibung der Return-Codes von SFC14, SFC15

FFFEYYYYh: Fehlermeldung vom Treiber

FFFE0001h: Ungültige Dlen bei Schreiben

FFFE0002h: Keine Antwort vom Slave (Timeout)

KKCCAAAAh: Fehler-Code vom Slave mit KKh: Fehler-Klasse

CCh: Fehler-Code

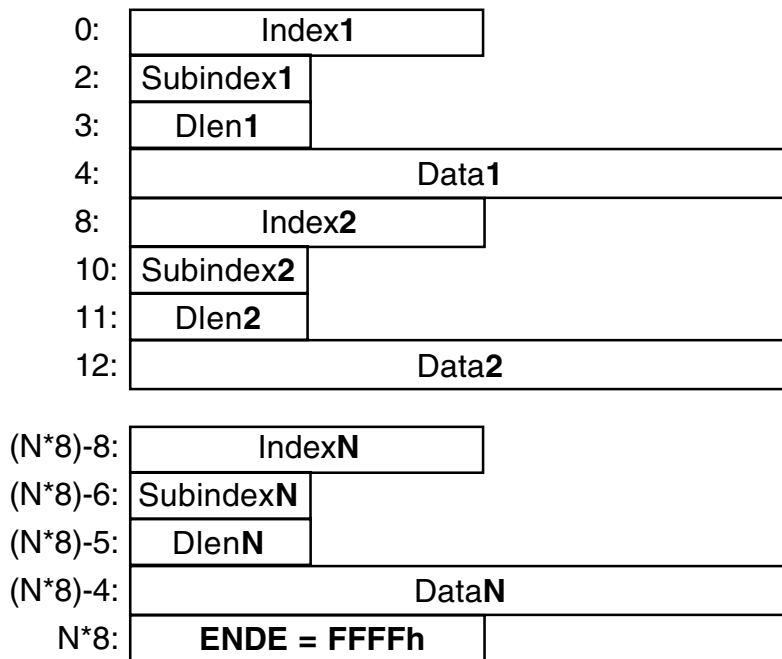
AAAAh: Additional-Code

<i>word PARAM_Ptr</i>	Parameter-Pointer:	Zeiger auf das 1. Peripheriebyte des Parametrierkanals dieses DP-Slaves
<i>byte Handshake</i>	Handshake-Merker:	Wird vom Treiber aktualisiert
<i>word Index</i>	Parameter-Index:	Adressierung des Parameters (Anwendungsprogramm)
<i>byte Subindex</i>	Parameter-Subindex:	Adressierung des Parameters (Anwendungsprogramm)
<i>byte Dlen</i>	Datenlänge in Byte:	Beim Schreiben von Anwendungsprogramm gesetzt, bei Lesen vom Treiber gesetzt
<i>dword Data</i>	Wert des Parameters:	Beim Schreiben von Anwendungsprogramm gesetzt, bei Lesen vom Treiber gesetzt
<i>word Time_Stamp</i>	Zeitstempel:	Wird vom Treiber nach Absenden der Parametrierkanal-Anforderung gesetzt und dient der Timeoutüberwachung.

0:	State
1:	Slave_Index
2:	Init_Index
3:	Slave_DBNR1
4:	Init_DBNR1
5:	Slave_DBNR2
6:	Init_DBNR2
(N*2)+1:	Slave_DBNRN
(N*2)+2:	Init_DBNRN
(N*2)+3:	ENDE = FFh

6.4 Aufbau des Init-DB

Der Init-DB enthält eine Liste von Schreibaufträgen, die während der Initialisierungsphase zu einem/mehreren Slaves geschrieben wird:



word Index1 Schreibauftrag 1, Index des Parameters.

byte Subindex1 Schreibauftrag 1, Subindex des Parameters.

byte Dlen1 Schreibauftrag 1, Datenlänge in Byte.

dword Data1 Schreibauftrag 1, Daten des Parameters.

word Index2 Schreibauftrag 2, Index des Parameters.

byte Subindex2 Schreibauftrag 2, Subindex des Parameters.

byte Dlen2 Schreibauftrag 2, Datenlänge in Byte.

dword Data2 Schreibauftrag 2, Daten des Parameters.

word IndexN Schreibauftrag N, Index des Parameters.

byte SubindexN Schreibauftrag N, Subindex des Parameters.

byte DlenN Schreibauftrag N, Datenlänge in Byte.

dword DataN Schreibauftrag N, Daten des Parameters.

word ENDE_KENNUNG FFFFh

6.5 Timeout-Überwachung

Der Parametrierkanal-Treiber enthält eine Timeout-Überwachung. Diese überwacht die Antwortzeit für Parametrierkanal-Aufträge. Übersteigt diese Antwortzeit einen gewissen Wert (standardmäßig ca. 1 s), beendet der Treiber den Auftrag mit der entsprechenden Fehlermeldung.

Realisiert ist die Timeoutüberwachung mittels eines zyklischen Timers (Zykluszeit = 100ms), der aus dem Verzögerungsalarm OB20 besteht. Dieser wird in der Initialisierung des Treibers (FC4, INIT_KEB) durch Aufruf von SFC32 gestartet. Dann startet sich der Verzögerungsalarm (OB20) immer wieder selbst. In dem Verzögerungsalarm wird eine globale Variable „TTICK“ erhöht. Wenn ein Parametrierkanal-Auftrag gestartet wurde, wird im Slave-DB unter der Zelle „Time_Stamp“ der aktuell Wert des „TTICK“ gesichert. Solange der Auftrag noch nicht bestätigt ist, wird die Differenz zwischen „Time-Stamp“ und dem aktuellen „TTICK“-Wert als Wartezeit herangezogen. In der vorliegenden Treiber-SW ist die Timeoutzeit fest programmiert auf 10 „TTICK“-Zyklen (siehe FC2, Netzwerk 4 und Netzwerk 5).

6.6 PROFIBUS-Diagnose

Man kann im S7-Anwendungsprogramm auf mehrfache Weise überprüfen, ob die projektierten Slaves alle korrekt arbeiten, oder ob irgendwo Probleme bestehen. Bei dem Parametrierkanal-Treiber ist dies jedoch nicht zusätzlich notwendig, da beim Zugriff auf den Parametrierkanal immer über die System-Funktionen SFC14 und SFC15 gearbeitet wird. Diese geben bei Auftreten eines Problems einen Fehler-Code zurück, so daß der Ausfall eines Slaves immer beim Zugriff auf seinen Parametrierkanal auffällt.

Dennoch ist in der Treiber-Software zusätzlich der Alarm-Baustein **OB86 (Baugruppenträgerausfall)** beispielhaft programmiert worden. Beim Ausfall eines projektierten Slave wird die PROFIBUS-Adresse dieses Slave auf Ausgangsbyte **AB0** ausgegeben. Weiterhin wird ein Zähler in **AB1** mitgeführt, der die Anzahl aufgetretener Ausfälle zählt.

An dieser Stelle soll nur noch hingewiesen werden auf den Alarm-Baustein **OB82 (Diagnosealarm)** und auf die Systemfunktion **SFC13 (DPNRM_DG)**.

7. Randbemerkungen

7.1 Konfiguration von KEB Slaves

In dem vorliegenden Beispielprojekt sind die beiden projektierten KEB Slaves unterschiedlich konfiguriert (s. **SIMATIC 300(1) -> Hardware**). Dies ist ganz bewußt so gemacht worden, um beide Möglichkeiten vorzustellen. Generell können bei der Konfiguration eines KEB Slaves die Modul-Definitionen aus der GSD-Datei übernommen (keb_dp1.gsd) oder aber von Hand über das sogenannte **Universalmodul** eingegeben werden. Die Vorgabe über das Universalmodul hat den Vorteil, daß Änderungen in der Slave-Konfiguration relativ einfach nachträglich vorgenommen werden können. Dies kann z.B. notwendig werden, wenn Sie dem/den KEB Slaves andere Prozessdatenbelegungen vorgegeben haben und sich dadurch die Konfiguration geändert hat. Die Vorgabe über die GSD-Datei mit den Modul-Namen **Parametrierkanal**, **Prozess-Ausgangsdaten** und **Prozess-Eingangsdaten** ist, wenn überhaupt, nur sehr umständlich möglich. D.h. sollten Sie in Ihrer Anwendung vorhaben, die Länge der Prozessdaten zu ändern, benutzen Sie bei der Konfiguration der KEB Slaves am besten für die beiden Prozessdaten-Module das **Universalmodul**. Wichtig ist dabei, daß bei der Konfiguration die **Einheit: „Byte“** und **Konsistent über: „gesamte Länge“** vorgegeben wird.

7.2 Zugriff auf konsistente Peripheriebereiche bei Simatic S7

Bitte beachten Sie beim Zugriff auf die Peripherie eines DP-Slaves, daß bei einer Länge von **drei** oder **mehr als vier Byte** der Zugriff ausschließlich über die System-Funktionen SFC14, SFC15 geschehen darf. In allen anderen Fällen ist der direkte Zugriff zu wählen. Aus diesem Grund ist in dem vorliegenden Treiber jeder Zugriff auf die Parametrierkanal-Peripherie über die o.g. Systemfunktionen realisiert worden.

8. Belegte P r o g r a m m - Ressourcen des Treibers

- Im Merkerbereich werden die Merker-Bytes 100 bis 125 benutzt (26 Merkerbytes). Diese dürfen von der Anwendungs-SW nicht verändert werden.
- Je projektiertem KEB Slave 1 Datenbaustein (DB) mit 18 Byte Länge.
- Je Initialisierungsblock wird ein Datenbaustein mit $(N \cdot 8) + 2$ Byte Länge benötigt, wobei N die Anzahl der Initialisierungsaufträge ist.
- 4 Funktionen (FC1, FC2, FC3, FC4). Die Länge der Funktionen entnehmen Sie bitte den Angaben zu den einzelnen Funktionen in den Objekteigenschaften im Simatic-Manager.
- 1 Verzögerungsalarm-OB (OB20). Dieser kann vom Anwendungsprogramm ebenfalls genutzt werden. Vom Parametrierkanal-Treiber wird er als zyklischer Timer mit einer Zykluszeit von 100ms betrieben und zur Timeout-Erkennung verwendet.
- 1 Baugruppenträgeralarm-OB (OB86). Dieser ist für die Funktion des Treibers nicht notwendig und kann ohne negative Auswirkungen gelöscht oder umprogrammiert werden.





KEB Austria
Ritzstraße 8 • A - 4614 Marchtrenk
Tel.: 0043 / 7243 / 53586 - 0 • FAX: 0043 / 7243 / 53586-21



KEBCO Inc.
1335 Mendota Heights Road
USA - Mendota Heights, MN 55120
Tel.: 001 / 651 / 4546162 • FAX: 001 / 651 / 4546198



KEB (UK) Ltd.
6 Chieftain Business Park, Morris Close
Park Farm, Wellingborough, GB - Northants, NN8 6 XF
Tel.: 0044 / 1933 / 402220 • FAX: 0044 / 1933 / 400724



KEB - YAMAKYU Ltd.
711 Fukudayama, Fukuda
J - Shinjo City, Yamagata (996-0053)
Tel.: 0081 / 233 / 29 / 2800 • FAX: 0081 / 233 / 29 / 2802



KEB Italia S.r.l.
Via Newton, 2 • I - 20019 SETTIMO MILANESE (Milano)
Tel.: 0039 / 02 / 33500782 • FAX: 0039 / 02 / 33500790



Société Française KEB
Z.I. de la Croix St Nicolas • 14, rue Gustave Eiffel
F - 94510 LA QUEUE EN BRIE
Tél.: 0033 / 1 / 49620101 • FAX: 0033 / 1 / 45767495



Karl E. Brinkmann GmbH
Försterweg 36 - 38 • D - 32683 Barntrup
Telefon 0 52 63 / 4 01 - 0 • Telefax 4 01 - 116
Internet: www.keb.de • E-mail: info@keb.de