

# Kurzanleitung/ Quick Guide



## Inhaltverzeichnis

Einleitung	Seite 2
Kontakt	Seite 2
Modus Lesen und Schreiben	Seite 3
Controlbyte	Seite 3
Datensatzauswahl	Seite 5
Beispiel	Seite 6
Modus Ringspeicher	Seite 17
Modus QDAS	Seite 18

## Index of contents

Introduction	page 2
Contact	page 2
Read and Write mode	page 3
Control Byte	page 3
Data set selection	page 5
Example	page 6
Ring buffer mode	page 17
QDAS mode	page 18

## Einleitung

HSDBASE wurde konzipiert um es für jeden Anwender möglich zu machen, Daten aus S7-Steuerungen mitzuschreiben. Auf die einfache Anwendbarkeit wurde bei der Entwicklung dieser Software das Hauptaugenmerk gelegt.

Zur schnellen Einarbeitung gibt es im Programm eine integrierte Popup-Hilfe, die direkt bei der Verwendung von HSDBASE jede Schaltfläche erklärt.

Alle grundlegenden Informationen zu HSDBASE und den Modi zum Datenloggen (*Zyklisch, Trigger, Änderung, Analyzer*) sind in der Popup-Hilfe bereits enthalten.

Diese Kurzanleitung beschreibt deshalb lediglich die Modi *Lesen und Schreiben, Ringspeicher*, sowie die Datenerfassung im *QDAS Datenformat*.

Weitere Informationen finden Sie auf unserer Homepage unter FAQ und HSDBASE Beispiele.

## Kontakt

Für noch offene Fragen sind wir jederzeit für Sie erreichbar.

## Introduction

HSDBASE was created to log data from S7 PLCs easily for every user. The simple applicability was in the focus of attention throughout development.

For fast familiarization a pop up help is integrated. It explains every button directly while using HSDBASE.

All general information for the data logging modes (*Cycle, Trigger, Change, Analyzer*) are already included in the pop up help. Therefore only the *Read and Write, Ring buffer* and *QDAS* mode is explained in the following.

Additional information is available on our website under FAQ and HSDBASE examples.

## Contact

If there are any other questions, please do not hesitate to contact us at any time.

HS Automation Software  
Im Öderich 2  
D-73663 Berglen  
Tel.: + 49 (0) 7195 - 588 639  
E-Mail: [info@hs-automation-software.de](mailto:info@hs-automation-software.de)  
Web: [www.hs-automation-software.de](http://www.hs-automation-software.de)

## Modus Lesen und Schreiben

**Hinweis:**

Dieser Modus ist nicht zum Datenloggen gedacht, sondern nur zum Austausch von Daten in beide Richtungen.

Verwenden Sie zum Datenloggen, besonders bei schnellen Aufzeichnungen, einen der anderen Modi, da hier deutlich höhere Datenraten erzielt werden können.

## Read and Write mode

**Advice:**

This mode shouldn't be used for data logging, but for the exchange of data in both directions.

For data logging, especially fast loggings, you should use one of the other modes. Clearly higher data rates can be achieved with them.

## Das Controlbyte

Alle Lese- und Schreibaufträge werden von der Steuerung angestoßen.

Sollen Daten geschrieben bzw. gelesen werden, muss die Steuerung das entsprechende Kommando in das Controlbyte schreiben. **HSDBASE** liest das Controlbyte zyklisch entsprechend der eingestellten Zeit. Wird ein Wert größer 100 gelesen, arbeitet **HSDBASE** dieses Kommando ab und schreibt anschließend als Quittierung einen Antwortcode (kleiner 100) in das Controlbyte.

## The Control Byte

All read and write jobs are activated through the PLC.

Should data be written or read, the PLC has to write the matching commando into the Control Byte. **HSDBASE** reads the Control Byte cyclic according to the adjusted time. If the value is bigger than 100, **HSDBASE** completes this commando and writes an answering code (smaller than 100) as confirmation into the Control Byte afterwards.

**Alle Codes im Überblick:****Survey of all codes:**

	<b>Kommandocodes</b>	<b>Commando codes</b>
101	Aus Datenbank in SPS (Datensatz bleibt erhalten) - z.B. Rezeptur laden	from database to PLC (dataset remains) - e.g. load recipe
102	Aus Datenbank in SPS (Datensatz wird gelöscht) - z.B. Auftrag laden	from database to PLC (dataset will be deleted) - e.g. load batch
103	Abfrage Datensatz vorhanden - z.B. für Abfrage ob überschrieben werden soll	query if dataset present - e.g. request before overwriting
111	Aus SPS in Datenbank (Überschreiben, wenn vorhanden) - z.B. Rezeptur speichern	from PLC to database (overwrite, if dataset exists) - e.g. save recipe
112	Aus SPS in Datenbank (Doppelt anlegen, wenn vorhanden) - z.B. Auftrag anlegen	from PLC to database (create doubly, if dataset exists) - e.g. create batch
113	Datensatz löschen - z.B. Rezept oder Auftrag löschen	delete dataset - e.g. delete recipe or batch
121	AnzeigeDB mit verfügbaren Rezepten/Aufträgen füllen	fill DisplayDB with available identifiers of recipes/batches
	<b>Antwortcodes</b>	<b>Answering codes</b>
1	Daten wurden in SPS geschrieben	data written to PLC
2	Daten wurden in SPS geschrieben und Datensatz wurde gelöscht	data written to PLC and dataset deleted
3	Datensatz wurde erstellt	dataset created
4	Datensatz wurde aktualisiert	dataset updated
5	Datensatz wurde gelöscht	dataset deleted
6	Datensatz vorhanden	dataset present
7	AnzeigeDB wurde gefüllt	DisplayDB was filled
	<b>Fehlercodes</b>	<b>Fault codes</b>
11	Unbekannter Befehl	unknown command
12	Fehler Datensatz nicht vorhanden	error dataset does not exist
13	Fehler beim Lesen aus Datenbank	error reading database/file
14	Fehler Datenbank/Tabelle/Datei nicht gefunden	error database/table/file not found
15	Fehler keine Kopfzeile vorhanden	error no headline present
16	Fehler Anzahl Spalten Kopfzeile falsch	error wrong number of columns (headline)
17	Fehler Anzahl Spalten Datenzeile falsch	error wrong number of columns (dataline)
18	Fehler beim Schreiben in Datenbank/Tabelle/Datei	error writing database/table/file
19	Fehler beim Konvertieren / Datensatz wurde nicht komplett geschrieben	error converting/dataset was not written complete
20	AnzeigeDB nicht vorhanden	DisplayDB not present
21	AnzeigeDB zu klein	DisplayDB to small
22	AnzeigeDB nicht angewählt	checkbox DisplayDB is unchecked

## Datensatzauswahl

In **HSDBASE** befindet sich in der Variablen-tabelle eine Spalte *Wert ändern*. Nur die angewählten Werte in dieser Spalte werden von **HSDBASE** in der Steuerung überschrieben. Die nicht angewählten Variablen werden zur Auswahl der Datensätze verwendet.

Um zu bestimmen, welcher Datensatz aus der Datenbank gelesen (bzw. aktualisiert) werden soll, muss die Auswahlvariable in **HSDBASE** unter den Variablen angelegt werden und das Häkchen in der Spalte *Wert ändern* muss **abgewählt** sein. Der Datensatz kann auch anhand einer Kombination von mehreren Variablen bestimmt werden. Vor jeder Schreib- / Leseoperation werden die Werte aus der Steuerung ausgelesen. Dann werden nur die Datensätze bearbeitet, bei denen die Auswahlvariablen in Steuerung und Datenbank übereinstimmen.

Werden in der Datenbank mehrere Datensätze ausgewählt, so werden alle diese Datensätze aktualisiert/gelöscht. Bei einem Lesebefehl werden die Werte des ersten ausgewählten Datensatzes verwendet.

Zum besseren Verständnis wird empfohlen sich das Beispiel in diesem Handbuch anzuschauen.

## Data set selection

Inside the variable table of **HSDBASE**, there is a column named *change value*. Only the selected values in this column are overwritten in the PLC. The unchecked variables are used for the selection of the data sets.

To select the dataset, that should be read (respectively actualized), the selection variable has to be created under the variables in **HSDBASE** and the column *change value* has to be **unchecked**. Data sets can be selected also through the combination of several variables. Before every write / read operation the values are read out of the PLC. Then only the data sets which have the same value (in the selection variables) in the PLC and the database are handled.

If you select several data sets, then all these data sets will be actualized / deleted. On a read command the values of the first selected data set will be used.

Please have a look on the example in this guide for better understanding.

**Beispiel**

Folgendes Beispiel zeigt eine einfache Rezeptur mit 3 Parametern. Dieses Beispiel lässt sich sehr einfach nachstellen und mit einer Variablen-tabelle in S7 können Sie relativ einfach alle Kommandos testen.

**Example**

The following example shows a simple recipe with 3 parameters. This example can be reproduced very easily and it is relatively simple to test all commands with a S7 variable table.

Legen Sie folgenden Datenbaustein an:

Please create the following data block:

Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	Controlbyte	BYTE	B#16#0	
+1.0	RecipeNr	BYTE	B#16#0	
+2.0	Parameter1	BOOL	FALSE	
+4.0	Parameter2	INT	0	
+6.0	Parameter3	REAL	0.000000e+00	
=10.0		END_STRUCT		

Bild1: Datenbaustein

Picture1: data block

Legen Sie in **HSDBASE** eine neue Tabelle an.  
Stellen Sie als Modus *Lesen und Schreiben* ein.  
Anschließend geben sie die Adresse des  
Controlbytes ein:

Create a new table in **HSDBASE**. Set *Read and Write*  
as mode and set the address of the  
Control Byte:

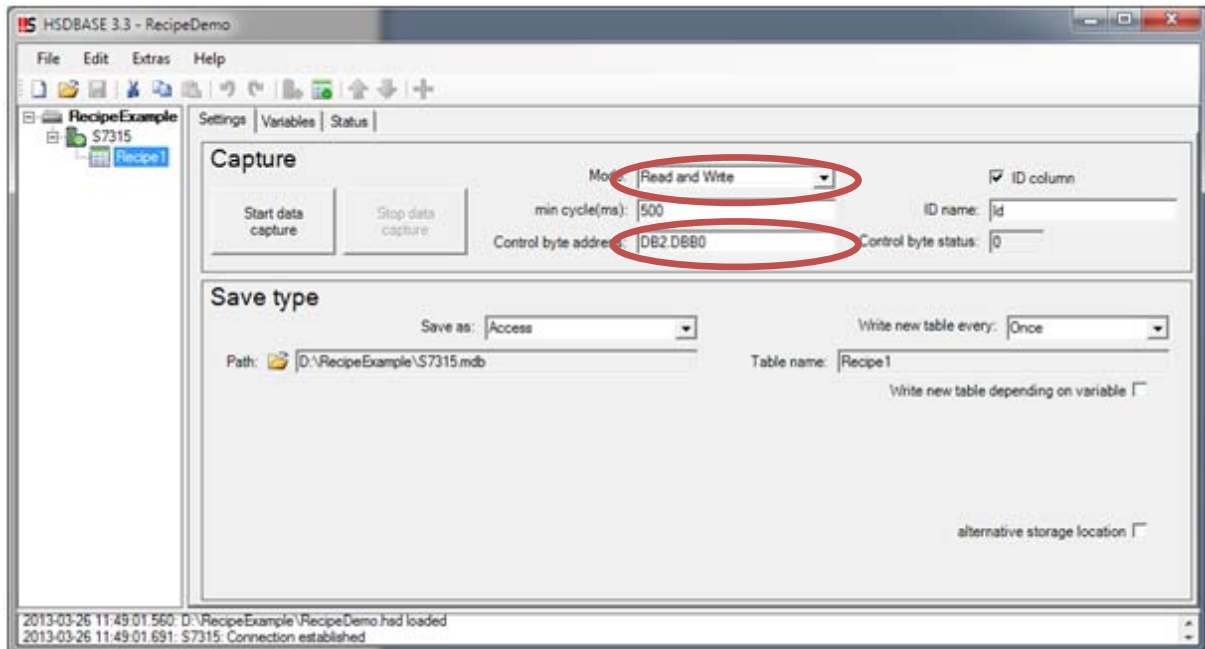


Bild2: Einstellungen **HSDBASE** (Settings)

Picture2: Settings **HSDBASE** (Settings)

Beim Anlegen der Variablen ist zu beachten, dass die Häkchen in der Spalte *Wert ändern* richtig gesetzt sind. Bei Rezeptnummer (Auswahlvariablen) nicht gesetzt, bei den Parametern gesetzt, da diese von **HSDBASE** entsprechend des Datenbankeintrags überschrieben werden:

Please note that the checkmarks in the column *change value* are placed correctly. Uncheck the recipe number (selection variables), but check the parameters because they are overwritten of **HSDBASE** according to the database entry:

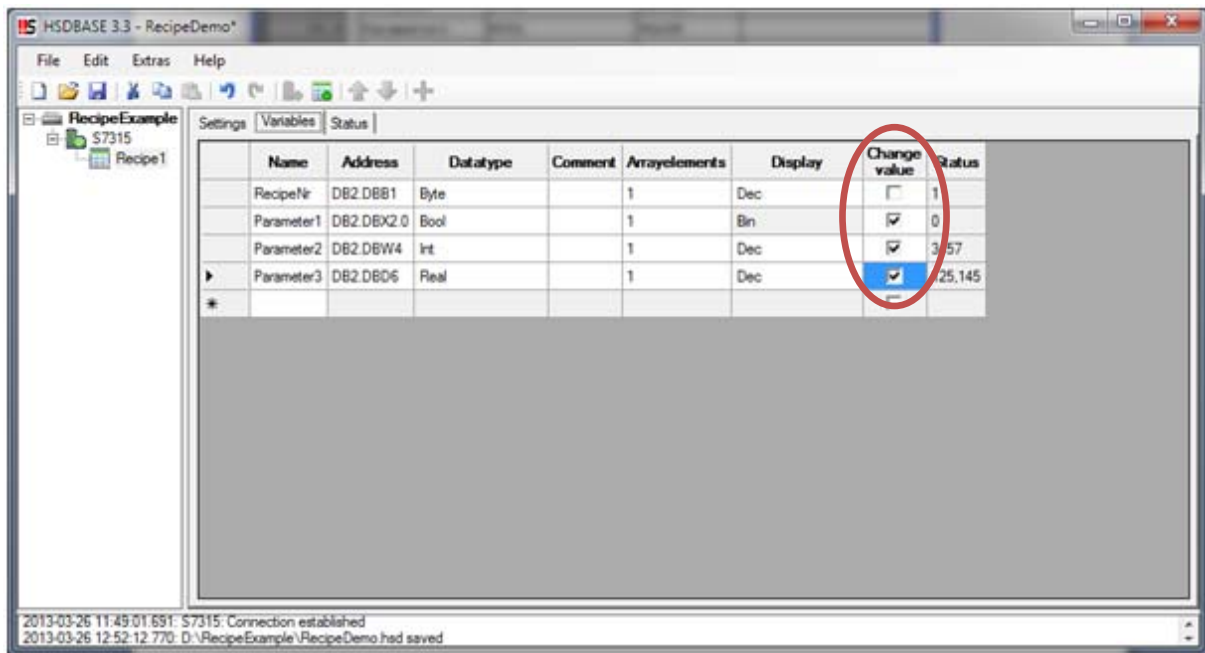


Bild3: Einstellungen **HSDBASE** (Variables)

Picture3: Settings **HSDBASE** (Variables)

Starten Sie nun die Erfassung, damit **HSDBASE** auf die Kommandos der SPS reagiert.

Start the capture now, so that **HSDBASE** can react to the comandos of the PLC.



**Rezeptur in Datenbank speichern**

Erstellen Sie eine Variablen-tabelle mit Rezeptnummer, Parameter und Control-byte. Stellen Sie für Rezeptnummer und Parameter Werte ein (Steuern):

**Save a recipe in the data base**

Create a variable table with recipe number, parameters and Control Byte. Set values for recipe number and parameters (control):

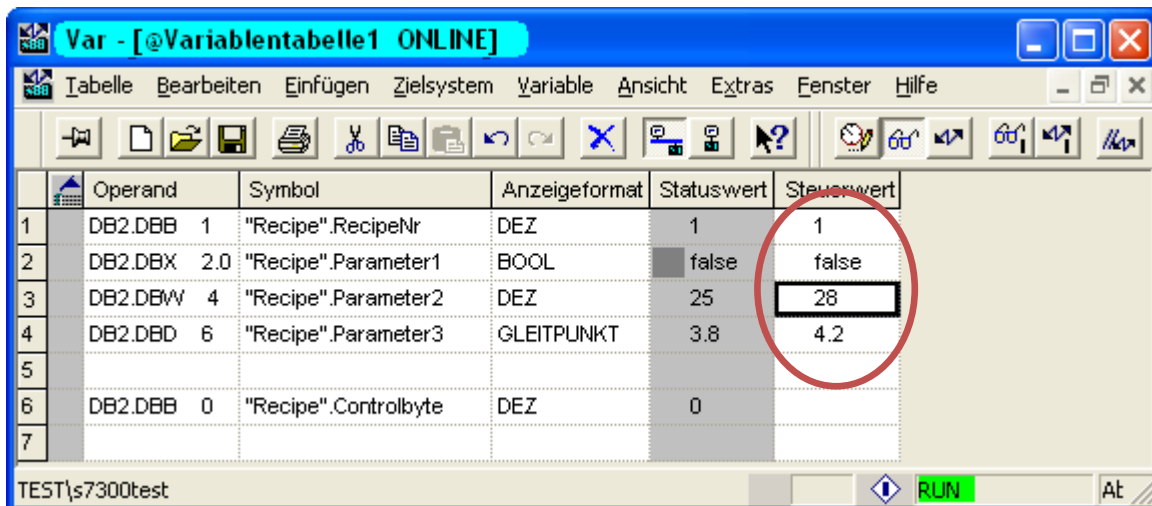


Bild1: Variablen-tabelle Werte setzen

Picture1: variable table set values

Steuern Sie nun das Controlbyte auf 111 (Aus SPS in Datenbank):

Set the Control Byte to 111 (From PLC to data base):

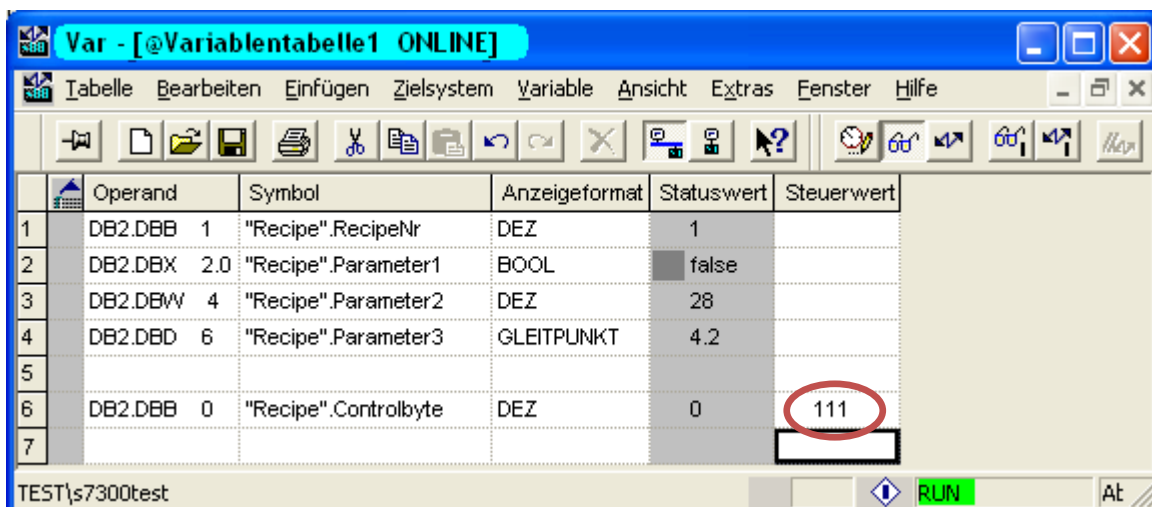


Bild2: Variablen-tabelle Kommandos setzen

Picture2: variable table set comandos

Als Antwortcode schreibt **HSDBASE** eine 3 (Daten-satz wurde erstellt) in das Controlbyte:

As answering code **HSDBASE** writes 3 (data set was created) into the Control Byte:

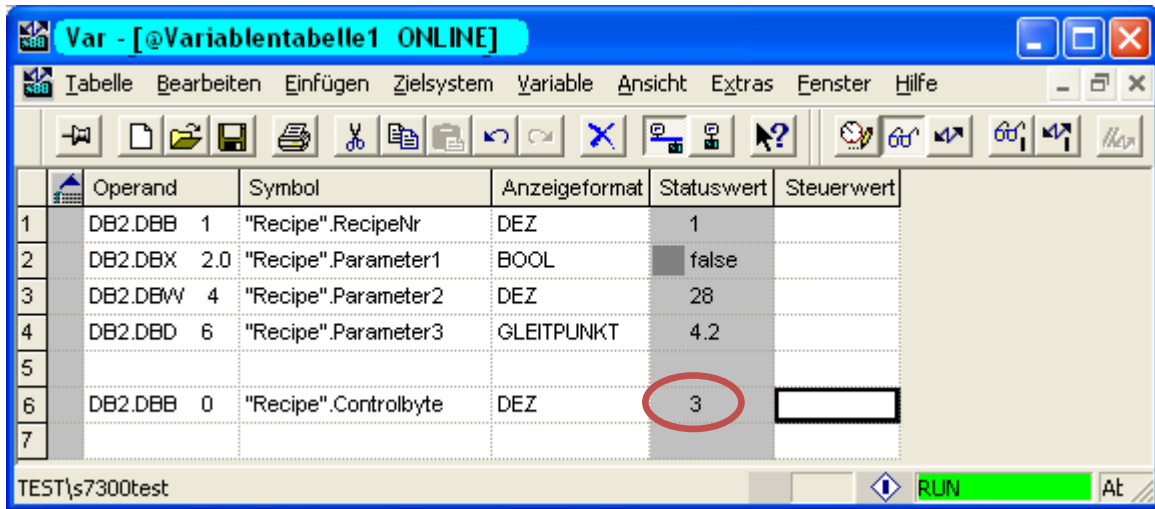


Bild3: Variablen-tabelle Antwortcode

Picture3: variable table answering code

Das Rezept befindet sich nun in der Datenbank. Datenbank und Tabelle wurde von **HSDBASE** automatisch angelegt:

Now the recipe is in the data base. Data base and table have been created automatically by **HSDBASE**:

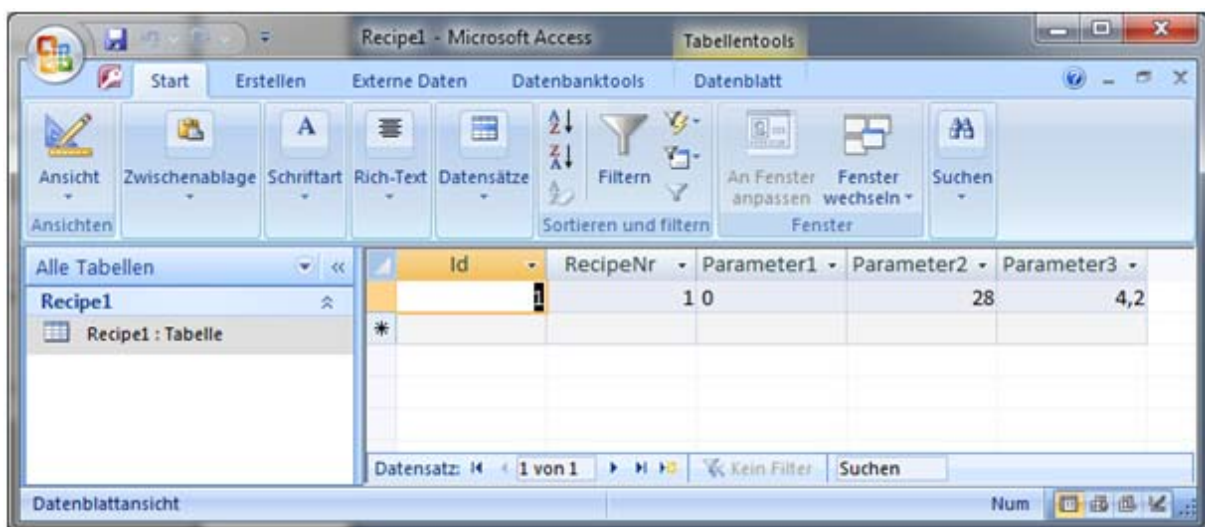
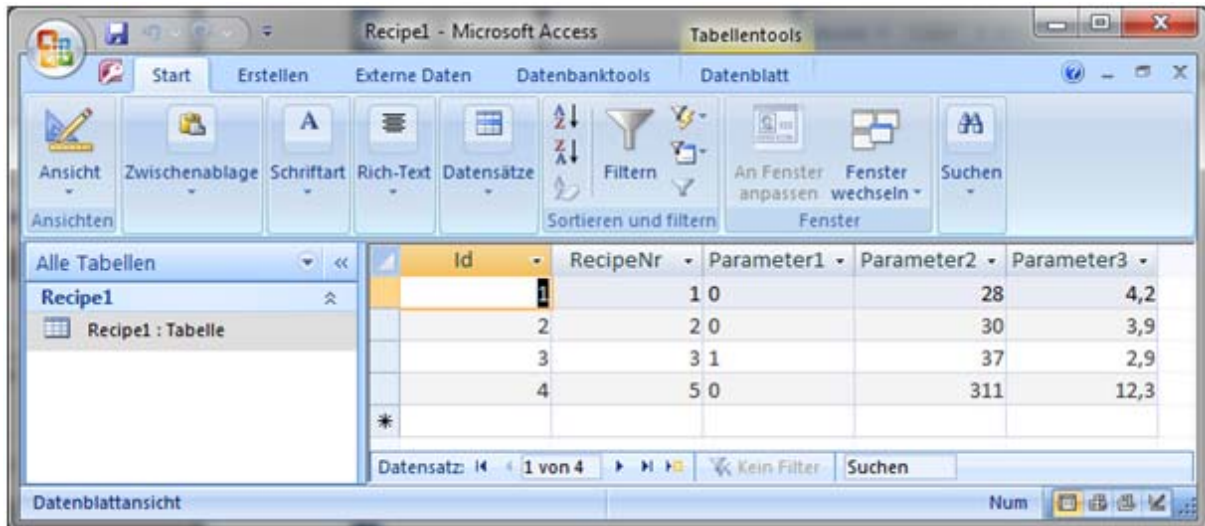


Bild4: Datensatz in Datenbank

Picture4: data set in the data base

Dieser Schritt kann nun beliebig oft mit verschiedenen Rezeptnummern und Werten wiederholt werden. Wird eine bereits verwendete Rezeptnummer benutzt, wird der Datensatz aktualisiert:

This step can be repeated as often as needed with different recipe numbers and values. If you apply an already used recipe number, the data set will be actualized:



The screenshot shows the Microsoft Access interface for a table named 'Recipe1'. The table contains four data records. The columns are 'Id', 'RecipeNr', 'Parameter1', 'Parameter2', and 'Parameter3'. The status bar at the bottom indicates 'Datensatz: 1 von 4' and 'Kein Filter'.

Id	RecipeNr	Parameter1	Parameter2	Parameter3
1	10		28	4,2
2	20		30	3,9
3	31		37	2,9
4	50		311	12,3

Bild5: Mehrere Datensätze in Datenbank

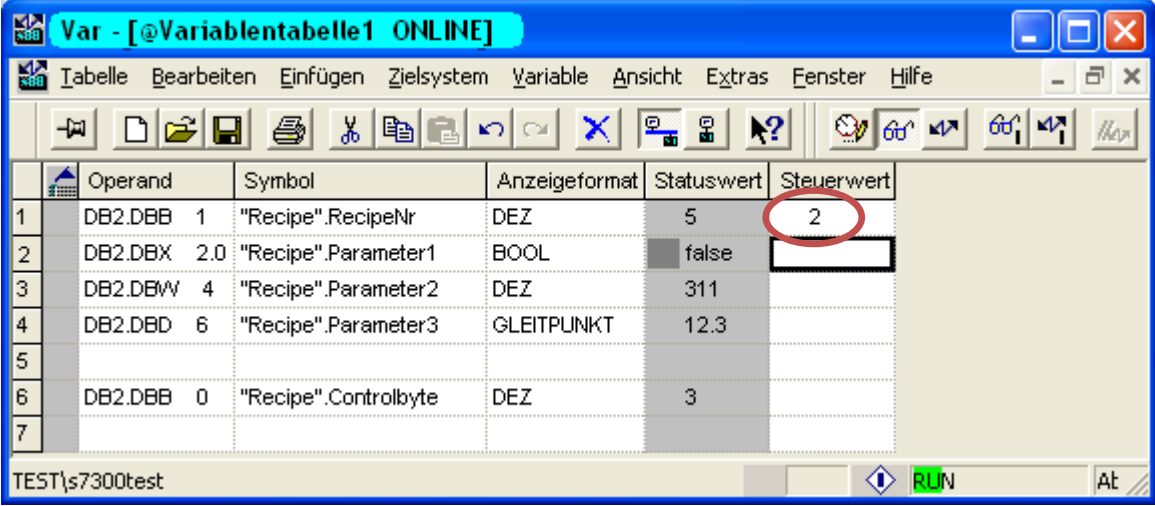
Picture5: several data sets in the data base

**Rezeptur aus Datenbank lesen.**

Nun soll das Rezept mit der Nummer 2 in die SPS übertragen werden. Hierzu muss die Rezeptnummer in der Steuerung eingestellt werden:

**Read recipe from data base**

Now the recipe with the number 2 should be transferred to the PLC. Therefore the recipe number has to be set first:



	Operand	Symbol	Anzeigeformat	Statuswert	Steuerwert
1	DB2.DBB 1	"Recipe".RecipeNr	DEZ	5	2
2	DB2.DBX 2.0	"Recipe".Parameter1	BOOL	false	
3	DB2.DBW 4	"Recipe".Parameter2	DEZ	311	
4	DB2.DBD 6	"Recipe".Parameter3	GLEITPUNKT	12.3	
5					
6	DB2.DBB 0	"Recipe".Controlbyte	DEZ	3	
7					

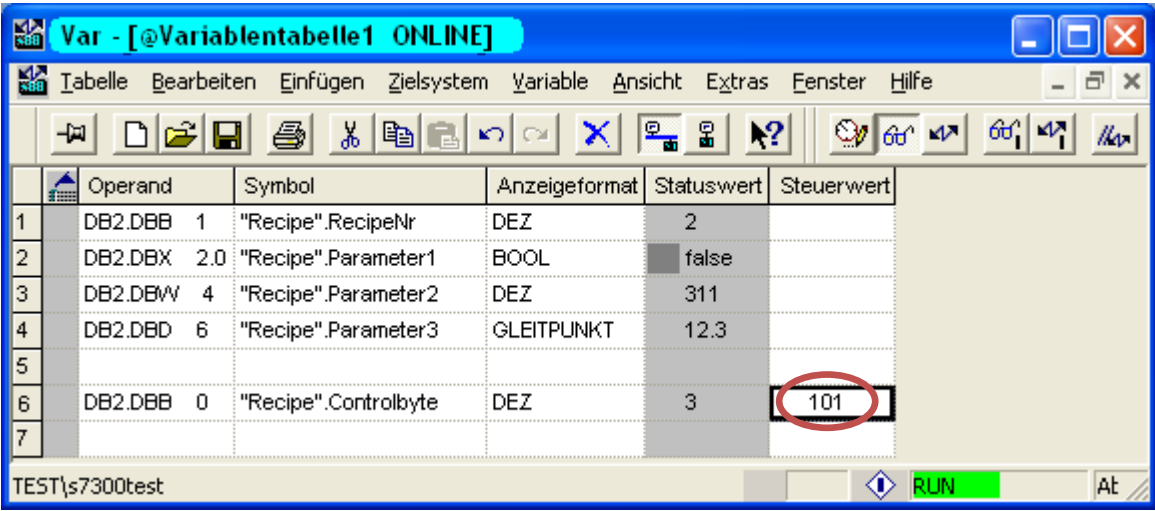
TEST\s7300test RUN At

Bild1: Variablentabelle Rezepturnummer setzen

Picture1: variable table set recipe number

Dann das Controlbyte auf 101 (Aus Datenbank in SPS) steuern:

Then set the Control Byte to 101 (from data base to PLC):



	Operand	Symbol	Anzeigeformat	Statuswert	Steuerwert
1	DB2.DBB 1	"Recipe".RecipeNr	DEZ	2	
2	DB2.DBX 2.0	"Recipe".Parameter1	BOOL	false	
3	DB2.DBW 4	"Recipe".Parameter2	DEZ	311	
4	DB2.DBD 6	"Recipe".Parameter3	GLEITPUNKT	12.3	
5					
6	DB2.DBB 0	"Recipe".Controlbyte	DEZ	3	101
7					

TEST\s7300test RUN At

Bild2: Variablentabelle Kommandos setzen

Picture2: variable table set commands

Die Parameter werden überschrieben und der Antwortcode 1 (Daten wurden in SPS geschrieben) gesetzt:

The parameters are overwritten and the answering code 1 was set (data was written in PLC):

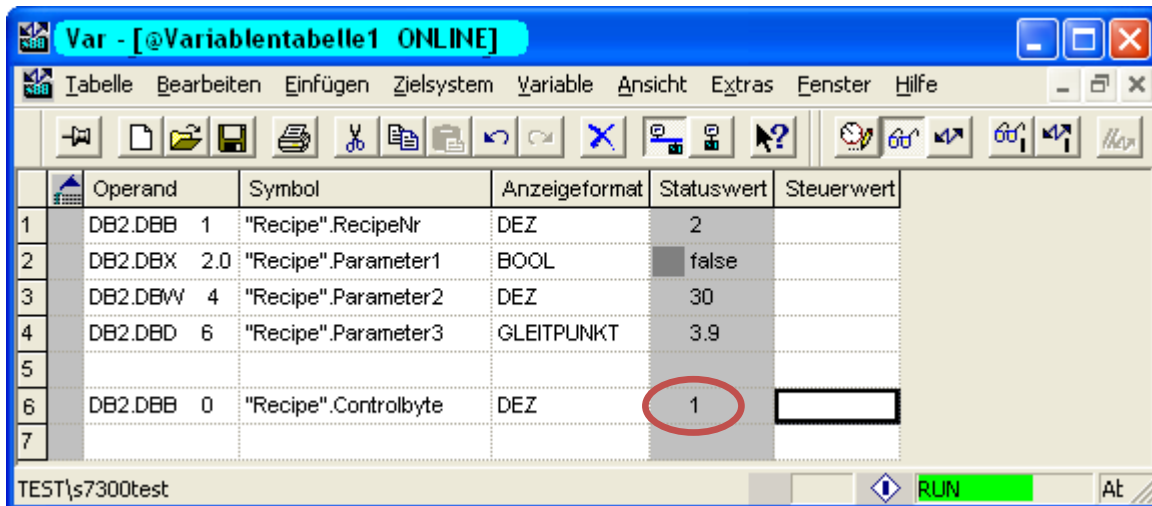


Bild3: Variablentabelle Antwortcode

Picture3: variable table answering code

**Rezeptur löschen**

Nun soll dasselbe Rezept (Nr. 2) gelöscht werden. Hierzu muss das Controlbyte auf 113 (Datensatz löschen) gesetzt werden:

**Delete recipe**

Now the same recipe (no. 2) should be deleted. Therefore the Control Byte has to be set to 113 (delete data set):

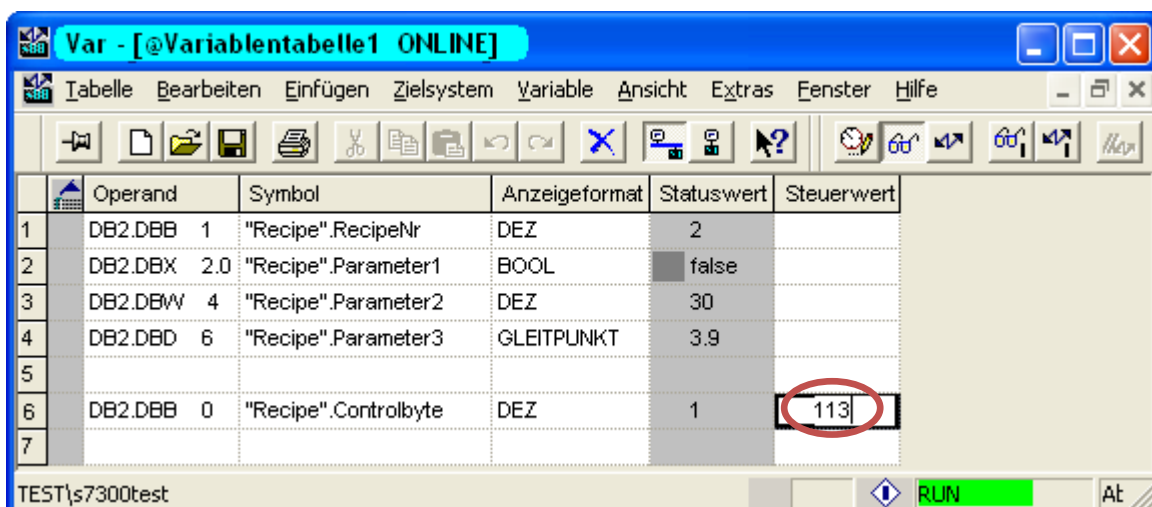


Bild1: Variablentabelle Kommandos setzen

Picture1: variable table set commandos

Als Antwortcode erhalten Sie 5 (Datensatz wurde gelöscht):

The answering code is 5 (data set was deleted):

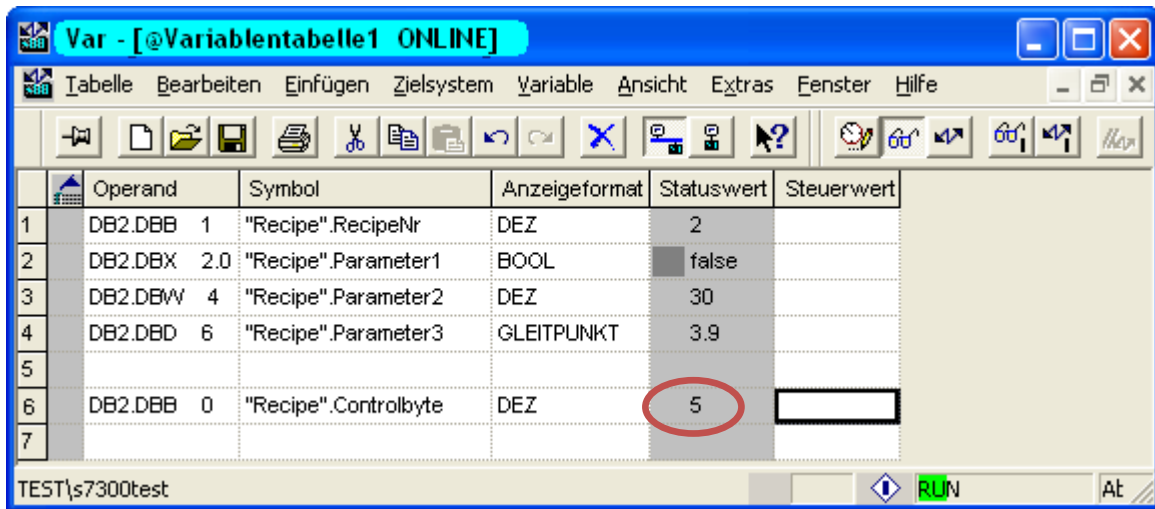


Bild2: Variablen-tabelle Antwortcode

Picture2: variable table answering code

Die Tabelle in der Datenbank sieht nun wie folgt aus:

Now the table looks as follows:

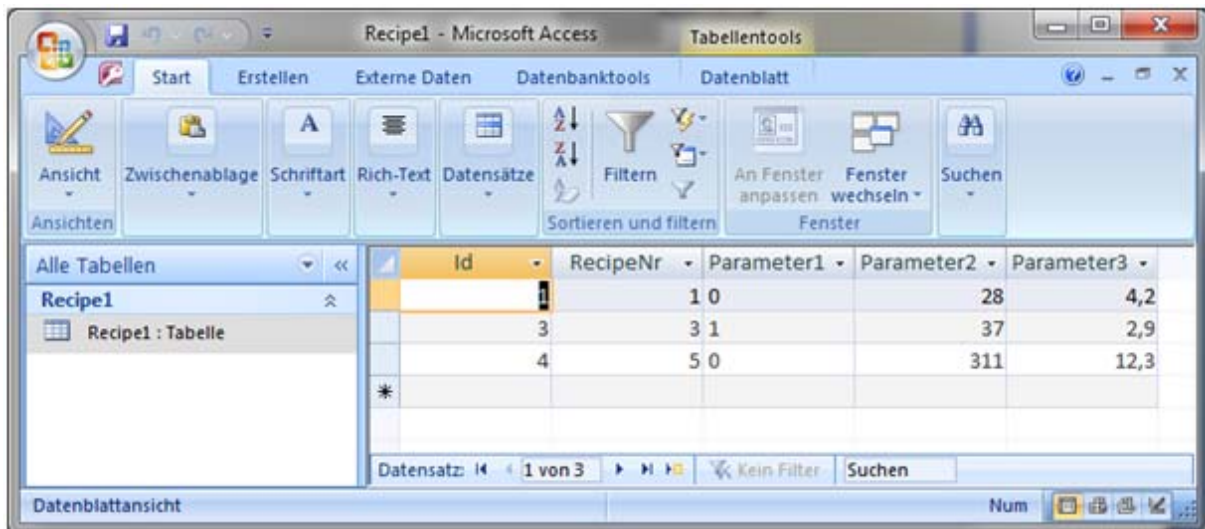


Bild3: Datensatz gelöscht

Picture3: data set was deleted



## Anzeige Datenbaustein

Um auszulesen welche Rezepturen in der Datenbank vorhanden sind, kann ein Anzeige-Datenbaustein erstellt werden. Dieser wird dann von **HSDBASE** beschrieben. Wird das Controlbyte auf 121 (AnzeigeDB füllen) gesetzt, schreibt **HSDBASE** die Anzahl aller Datensätze in der Datenbank in das erste Datendoppelwort. Im dritten Datendoppelwort steht die Anzahl der Datensätze, die in den Datenbaustein geschrieben wurden. Der Rest des DBs wird mit den Bezeichnern der Datensätze aufgefüllt. **HSDBASE** bestimmt automatisch wie viele Bezeichner in den Baustein passen. Sollten mehr Datensätze in der Datenbank vorhanden sein, als in den DB passen, kann der Beginn des Anzeigebereichs über das zweite Datendoppelwort vorgegeben werden. Sind weniger Datensätze vorhanden, wird der restliche DB mit w#16#00 gefüllt.

## Display data block

To read out which recipes are present in the database, a data block can be created. This data block will be filled by **HSDBASE**, when the Controlbyte is set to 121 (fill DisplayDB). **HSDBASE** writes the number of all datasets in the database into the first DINT. The third DINT contains the number of datasets, which have been written into the data block. The rest of the data block is filled up with the identifiers of the data sets. How many identifiers match into the block detects **HSDBASE** automatically. If more data sets exist in the data base than space is available, the beginning of the display area can be adjusted with the second DINT. If there are less datasets than space, the remaining space will be filled with w#16#00.

 <p>Bild1: Aufbau Anzeige DB</p>
 </div>
 <div data-bbox="521 773 770 790" data-label="Caption">
 <p>Picture1: structure of DisplayDB</p>
 </div>
 <div data-bbox="113 865 473 900" data-label="Text">
 <p>Alle weiteren Kommandos können auf diese Weise getestet werden. Bei der Erstellung des</p>
 </div>
 <div data-bbox="521 865 888 900" data-label="Text">
 <p>SPS-Programms sollte darauf geachtet werden, dass die Werte vor dem Controlbyte</p>
 </div>
 <div data-bbox="121 907 426 924" data-label="Page-Footer">
 <p>© HS Automation Software 2018-02-05</p>
 </div>
 <div data-bbox="747 907 878 924" data-label="Page-Footer">
 <p>Seite 15 von 18</p>
 </div>

gesetzt werden. Ebenso ist zu beachten, dass das Controlbyte nicht dauerhaft gesetzt wird.

All other commandos can be tested in this way. Please note, when creating the PLC program, the values have to be set before the Control Byte. Also note that the Control Byte isn't set permanently.



## Modus Ringspeicher

In diesem Modus werden die Daten in RAM des Rechners, auf dem **HSDBASE** läuft gesammelt und erst auf Befehl in die Datenbank geschrieben. Der Ringspeicher wird wie im Modus Lesen und Schreiben über ein Controlbyte gesteuert.

## Mode ring buffer

In this mode, data is collected in a FIFO buffer on the machine that runs **HSDBASE**. Data is written to the database on command. The buffer is controlled by a Control Byte like the Read and Write mode.

### Alle Codes im Überblick:

### Survey of all codes:

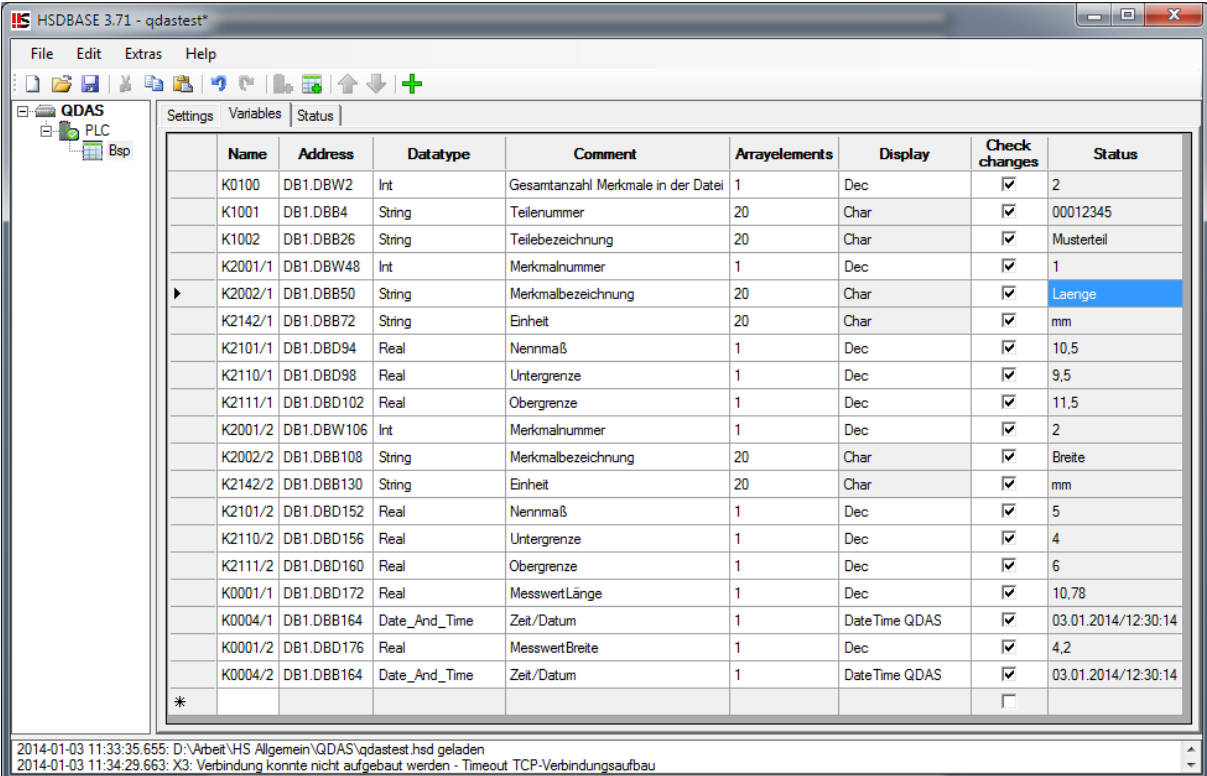
	<b>Kommandocodes</b>	<b>Commando codes</b>
101	Ringspeicher löschen	delete buffer
102	Ringspeicher füllen (wird nicht quittiert)	fill buffer (no acknowledgment)
103	Ringspeicher abspeichern	save buffer
	<b>Antwortcodes</b>	<b>Answering codes</b>
1	Ringspeicher wurde gelöscht	buffer deleted
2	Ringspeicher wurde gespeichert (und gelöscht)	buffer saved ( and deleted)
11	unbekannter Befehl	unknown command

## Modus QDAS

Der folgende Abschnitt zeigt anhand eines Beispiels, wie **HSDBASE** konfiguriert werden muss, um Daten im QDAS-Datenformat DFQ abzuspeichern. Diese können dann bspw. mithilfe des QDAS-Uploadtools automatisch in die QDAS-Datenbank eingelesen werden.

## QDAS mode

The following example shows how **HSDBASE** has to be configured to save the data as QDAS DFQ file. These files could be read by QDAS upload tool and transferred to the QDAS database automatically.



Name	Address	Datatype	Comment	Arrayelements	Display	Check changes	Status
K0100	DB1.DBW2	Int	Gesamtanzahl Merkmale in der Datei	1	Dec	<input checked="" type="checkbox"/>	2
K1001	DB1.DBB4	String	Teilenummer	20	Char	<input checked="" type="checkbox"/>	00012345
K1002	DB1.DBB26	String	Teilebezeichnung	20	Char	<input checked="" type="checkbox"/>	Musterteil
K2001/1	DB1.DBW48	Int	Merkmalnummer	1	Dec	<input checked="" type="checkbox"/>	1
K2002/1	DB1.DBB50	String	Merkmalbezeichnung	20	Char	<input checked="" type="checkbox"/>	Laenge
K2142/1	DB1.DBB72	String	Einheit	20	Char	<input checked="" type="checkbox"/>	mm
K2101/1	DB1.DBD94	Real	Nennmaß	1	Dec	<input checked="" type="checkbox"/>	10,5
K2110/1	DB1.DBD98	Real	Untergrenze	1	Dec	<input checked="" type="checkbox"/>	9,5
K2111/1	DB1.DBD102	Real	Obergrenze	1	Dec	<input checked="" type="checkbox"/>	11,5
K2001/2	DB1.DBW106	Int	Merkmalnummer	1	Dec	<input checked="" type="checkbox"/>	2
K2002/2	DB1.DBB108	String	Merkmalbezeichnung	20	Char	<input checked="" type="checkbox"/>	Breite
K2142/2	DB1.DBB130	String	Einheit	20	Char	<input checked="" type="checkbox"/>	mm
K2101/2	DB1.DBD152	Real	Nennmaß	1	Dec	<input checked="" type="checkbox"/>	5
K2110/2	DB1.DBD156	Real	Untergrenze	1	Dec	<input checked="" type="checkbox"/>	4
K2111/2	DB1.DBD160	Real	Obergrenze	1	Dec	<input checked="" type="checkbox"/>	6
K0001/1	DB1.DBD172	Real	MesswertLänge	1	Dec	<input checked="" type="checkbox"/>	10,78
K0004/1	DB1.DBB164	Date_And_Time	Zeit/Datum	1	DateTime QDAS	<input checked="" type="checkbox"/>	03.01.2014/12:30:14
K0001/2	DB1.DBD176	Real	MesswertBreite	1	Dec	<input checked="" type="checkbox"/>	4,2
K0004/2	DB1.DBB164	Date_And_Time	Zeit/Datum	1	DateTime QDAS	<input checked="" type="checkbox"/>	03.01.2014/12:30:14
*						<input type="checkbox"/>	

2014-01-03 11:33:35.655: D:\Arbeit\HS\_Allgemein\QDAS\qdastest.hsd geladen  
 2014-01-03 11:34:29.663: X3: Verbindung konnte nicht aufgebaut werden - Timeout TCP-Verbindungsaufbau

Bild3: Einstellungen **HSDBASE** (Variables)

Picture3: Settings **HSDBASE** (Variables)

Die für QDAS benötigten K-Nummern sind in der Spalte Name einzutragen. Pro Trigger wird eine separate DFQ-Datei mit fortlaufender Nummer generiert.

The QDAS K numbers have to be entered in the name column. A separate DFQ file with consecutive number will be created on every trigger event.